aws

Leitfaden

AWS OpsWorks

API-Version 2013-02-18

AWS OpsWorks: Leitfaden

Table of Contents

Was ist AWS OpsWorks?

> **Important**
>
> Die AWS OpsWorks Dienste haben das Ende ihrer Lebensdauer erreicht und wurden sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks ist ein Konfigurationsverwaltungsdienst, der Sie bei der Konfiguration und dem Betrieb von Anwendungen in einem Cloud-Unternehmen mithilfe von Puppet oder Chef unterstützt. AWS OpsWorks Stacks ermöglicht es Ihnen, Chef-Kochbücher und -Lösungen für das Konfigurationsmanagement zu verwenden, während OpsWorks Sie bei Puppet Enterprise einen Puppet Enterprise-Masterserver konfigurieren können. AWS OpsWorks for Chef Automate AWS Puppet bietet eine Reihe von Tools für die Erzwingung des gewünschten Zustands Ihrer Infrastruktur und die Automatisierung der On-Demand-Aufgaben.

AWS OpsWorks Dienste

AWS OpsWorks für Puppet Enterprise

OpsWorks für Puppet Enterprise können Sie AWS verwaltete Puppet-Masterserver erstellen. Ein Puppet Master-Server verwaltet Knoten in Ihrer Infrastruktur, speichert Fakten über diese Knoten und dient als zentrales Repository für Ihre Puppet-Module. Module sind wiederverwendbare und gemeinsam nutzbare Einheiten von Puppet-Code, die Anweisungen dazu enthalten, wie Ihre Infrastruktur konfiguriert werden soll. Sie können Community-Module aus dem Puppet Forge herunterladen oder das Puppet Development Kit verwenden, um eigene angepasste Module zu erstellen, und dann ihre Bereitstellung mit dem Puppet Code Manager verwalten.

OpsWorks for Puppet Enterprise bietet einen vollständig verwalteten Puppet-Master, eine Suite von Automatisierungstools, mit denen Sie Ihre Anwendungen überprüfen, bereitstellen, betreiben und zukunftssicher machen können, sowie Zugriff auf eine Benutzeroberfläche, über die Sie Informationen über Ihre Knoten und Puppet-Aktivitäten einsehen können.

OpsWorks for Puppet Enterprise ermöglicht es Ihnen, mithilfe von Puppet zu automatisieren, wie Knoten konfiguriert, bereitgestellt und verwaltet werden, unabhängig davon, ob es sich um Amazon EC2 EC2-Instances oder lokale Geräte handelt. Ein Master OpsWorks für Puppet Enterprise bietet umfassende Automatisierung, indem er Aufgaben wie Software- und Betriebssystemkonfigurationen, Paketinstallationen, Datenbankeinrichtungen, Änderungsmanagement, Durchsetzung von Richtlinien, Überwachung und Qualitätssicherung übernimmt.

Da OpsWorks für Puppet Enterprise die Puppet Enterprise-Software verwaltet wird, kann Ihr Server zu einem von Ihnen gewünschten Zeitpunkt automatisch gesichert werden, er läuft immer mit der aktuellsten AWS-kompatiblen Version von Puppet und es werden immer die aktuellsten Sicherheitsupdates angewendet. Sie können Amazon EC2 Auto Scaling Scaling-Gruppen verwenden, um Ihrem Server automatisch neue Amazon EC2 EC2-Knoten zuzuordnen.

AWS OpsWorks für Chef Automate

AWS OpsWorks for Chef Automate ermöglicht es Ihnen, AWS verwaltete Chef-Server zu erstellen, die die Premium-Funktionen von Chef Automate enthalten, und diese mit dem Chef DK und anderen Chef-Tools zu verwalten. Ein Chef-Server verwaltet Knoten in Ihrer Umgebung, speichert Informationen über diese Knoten und dient als zentrales Repository für Ihre Chef-Rezeptbücher. Die Rezeptbücher enthalten Rezepte, die vom Chef Infra-(`chef-client`)-Agenten auf jedem mithilfe von Chef verwalteten Knoten ausgeführt werden. Sie können Chef-Tools wie `knife`und Test Kitchen verwenden, um Knoten und Kochbücher auf einem Chef-Server im Service zu verwalten. AWS OpsWorks for Chef Automate

Chef Automate ist ein im Lieferumfang enthaltenes Server-Softwarepaket, das einen automatisierten Workflow für kontinuierliche Bereitstellungs- und Konformitätsprüfungen bietet. AWS OpsWorks for Chef Automate installiert und verwaltet Chef Automate, Chef Infra und Chef InSpec mithilfe einer einzigen Amazon Elastic Compute Cloud-Instanz. Mit AWS OpsWorks for Chef Automate können Sie von der Community verfasste oder benutzerdefinierte Chef-Kochbücher verwenden, ohne spezifische Änderungen vornehmen zu müssen. AWS OpsWorks

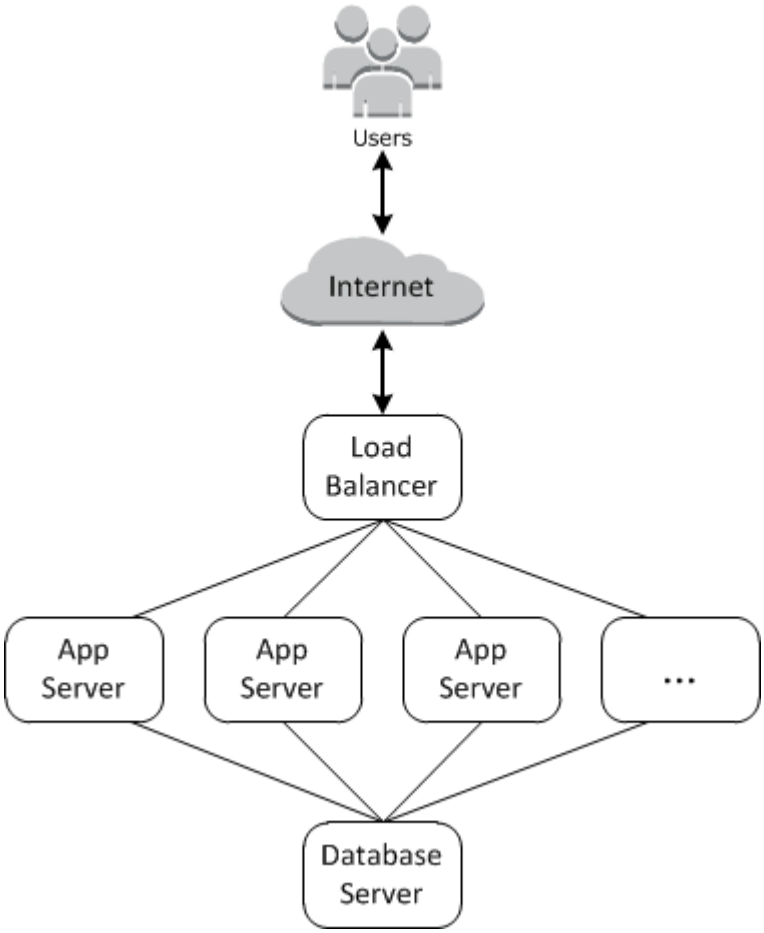
Da Chef Automate-Komponenten auf einer einzigen Instanz AWS OpsWorks for Chef Automate verwaltet werden, kann Ihr Server zu einem von Ihnen gewählten Zeitpunkt automatisch gesichert werden, auf dem immer die aktuellste Nebenversion von Chef ausgeführt wird und es werden immer die aktuellsten Sicherheitsupdates angewendet. Sie können Amazon EC2 Auto Scaling Scaling-Gruppen verwenden, um Ihrem Server automatisch neue Amazon EC2 EC2-Knoten zuzuordnen.

AWS OpsWorks Stapel

Cloud-basiertes Computing umfasst in der Regel Gruppen von AWS-Ressourcen, wie EC2-Instances und Amazon Relational Database Service (RDS) -Instances. Beispielsweise sind für eine Webanwendung normalerweise ein Anwendungsserver, Datenbankserver, Load Balancer und andere Ressourcen erforderlich. Diese Gruppe von Instances wird üblicherweise als Stack bezeichnet.

AWS OpsWorks Stacks, der ursprüngliche Service, bietet eine einfache und flexible Möglichkeit, Stacks und Anwendungen zu erstellen und zu verwalten. AWS OpsWorks Mit Stacks können Sie Anwendungen in Ihren Stacks bereitstellen und überwachen. Sie können die Stacks erstellen, mit deren Hilfe Sie Cloud-Ressourcen in spezialisierten Gruppen, die auch Ebenen genannt werden, verwalten können. Eine Ebene stellt eine Gruppe von EC2-Instances dar, die einem bestimmten Zweck dienen, wie die Bedienung von Anwendungen oder das Hosten von Datenbankservern. Die Ebenen sind von Chef-Rezepten zum Verarbeiten von Aufgaben, wie z. B. Installation von Paketen auf Instances, Bereitstellung von Anwendungen und Ausführen von Skripts, abhängig.

Im AWS OpsWorks for Chef Automate Gegensatz AWS OpsWorks dazu benötigt oder erstellt Stacks keine Chef-Server. AWS OpsWorks Stacks erledigt einen Teil der Arbeit eines Chef-Servers für Sie. AWS OpsWorks Stacks überwacht den Zustand der Instances und stellt bei Bedarf mithilfe von Auto Healing und Auto Scaling neue Instances für Sie bereit. Ein Beispiel für einen einfachen Anwendungsserver-Stack ist in folgender Abbildung dargestellt.


Users
Internet
Load Balancer
App Server
App Server
App Server
...
Database Server

AWS OpsWorks für Puppet Enterprise

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

OpsWorks für Puppet Enterprise können Sie innerhalb von Minuten einen Puppet Enterprise-Master starten und dessen Betrieb, Backups, Wiederherstellungen und Software-Upgrades AWS OpsWorks übernehmen. OpsWorks für Puppet Enterprise können Sie sich auf die wichtigsten Aufgaben des Konfigurationsmanagements konzentrieren, anstatt einen Puppet-Master zu verwalten. Durch OpsWorks die Verwendung von for Puppet Enterprise können Sie dieselben Konfigurationen verwenden, um sowohl Ihre lokale als auch Ihre Cloud-Infrastruktur zu verwalten, sodass Sie Ihre Abläufe in einer Hybridumgebung effizient skalieren können. Die Verwaltung des Puppet-Master-Servers wird durch die Puppet Enterprise-Konsole, die AWS Management Console und die AWS CLI vereinfacht.

Ein Puppet-Master verwaltet die Konfiguration von Knoten in Ihrer Umgebung, indem Konfigurationskataloge für bestimmte Knoten für die `puppet-agent`-Software bereitstellt werden, und dient als zentrales Repository für Ihre Puppet-Module. Ein Puppet-Master OpsWorks für Puppet Enterprise wird `puppet-agent` auf Ihren verwalteten Knoten bereitgestellt und bietet Premium-Funktionen von Puppet Enterprise.

Ein OpsWorks for Puppet Enterprise Master läuft auf einer Amazon Elastic Compute Cloud-Instanz. OpsWorks für Puppet Enterprise sind Server so konfiguriert, dass sie die neueste Version von Amazon Linux (Amazon Linux 2) und die aktuelle Version von Puppet Enterprise Master, Version 2019.8.5, ausführen. Weitere Informationen zu den Änderungen in Puppet Enterprise 2019.8.5 finden Sie in den Versionshinweisen zu Puppet Enterprise.

Wenn neue Versionen der Puppet-Software verfügbar werden, aktualisiert die Systemwartung die Version von Puppet Enterprise auf dem Server automatisch, wenn sie den AWS-Test bestanden hat. AWS führt umfangreiche Tests durch, um sicherzustellen, dass Puppet-Upgrades für die Produktion bereit sind und bestehende Kundenumgebungen nicht stören.

Sie können jeden lokalen Computer oder jede EC2-Instanz, auf der ein unterstütztes Betriebssystem ausgeführt wird und Netzwerkzugriff hat, mit einem for Puppet Enterprise Master verbinden. OpsWorks Die `puppet`-Agent-Software wird vom Puppet-Master auf Knoten installiert, die Sie verwalten möchten.

Themen

Regionalunterstützung OpsWorks für Puppet Enterprise

Die folgenden regionalen Endgeräte unterstützen Puppet OpsWorks Enterprise-Master. OpsWorks for Puppet Enterprise erstellt Ressourcen, die Ihren Puppet-Mastern zugeordnet sind, wie Instanzprofile, Benutzer und Servicerollen, auf demselben regionalen Endpunkt wie Ihr Puppet-Master. Ihr Puppet-Master muss sich in einer VPC befinden. Sie können eine VPC verwenden, die Sie erstellt oder bereits zur Verfügung haben, oder Sie verwenden die Standard-VPC.

- Region USA Ost (Ohio)
- Region USA Ost (Nord-Virginia)

- Region US West (N. California)
- Region USA West (Oregon)
- Region Asien-Pazifik (Tokio)
- Region Asien-Pazifik (Singapur)
- Region Asien-Pazifik (Sydney)
- Region Europa (Frankfurt)
- Europe (Ireland) Region

AWS OpsWorks for Puppet Enterprise Häufig gestellte Fragen zum Lebensende

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Themen

Wie werden Bestandskunden von diesem Lebensende betroffen sein?

Bestandskunden sind bis zum 31. März 2024, dem Enddatum OpsWorks für Puppet Enterprise, nicht betroffen. Nach dem Ende des Lebenszyklus können Kunden ihre Server nicht mehr über die OpsWorks Konsole oder API verwalten.

Was passiert mit meinen Servern, wenn ich nichts unternehme?

Ab dem 31. März 2024 können Sie Ihre Server nicht mehr über die OpsWorks Konsole oder API verwalten. Zu diesem Zeitpunkt werden wir die Durchführung aller laufenden Verwaltungsfunktionen für Ihre Server wie Backups oder Wartungsarbeiten einstellen. Um die Auswirkungen auf die Kunden zu begrenzen, lassen wir die EC2-Instances laufen, die Puppet Enterprise-Server sichern. Ihre Lizenzen sind jedoch nicht mehr gültig, da die Nutzung nicht mehr im Rahmen des Servicevertrags OpsWorks für Puppet Enterprise abgedeckt (oder in Rechnung gestellt) ist. Wenn Sie Ihre Infrastruktur weiterhin mit Puppet Enterprise verwalten möchten, finden Sie weitere Informationen unter So migrieren Sie einen OpsWorks for Puppet Enterprise-Server zu Amazon Elastic Compute Cloud (Amazon EC2).

Nimmt er neue Kunden an AWS OpsWorks for Puppet Enterprise ?

Nein. AWS OpsWorks for Puppet Enterprise akzeptiert keine neuen Kunden mehr.

Wird sich das Lebensende auf alle AWS-Regionen gleichzeitig auswirken?

Ja. Die API und die Konsole erreichen das Ende ihrer Nutzungsdauer und können ab dem 31. März 2024 in allen Regionen nicht mehr verwendet werden. Eine Liste der AWS-Regionen verfügbaren Dienste finden Sie unter Liste der AWS regionalen Dienste. AWS OpsWorks for Puppet Enterprise

Für welches Maß an technischem Support steht es zur Verfügung AWS OpsWorks for Puppet Enterprise?

AWS wird bis zum Ende der Nutzungsdauer weiterhin AWS OpsWorks for Puppet Enterprise das gleiche Maß an Support bieten, das Kunden heute haben. Wenn Sie Fragen oder Bedenken haben, können Sie das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support kontaktieren.

Ich bin ein aktueller Kunde von OpsWorks Puppet Enterprise und muss einen Server in einem Konto starten, das den Dienst zuvor nicht genutzt hat. Bin ich dazu in der Lage?

Im Allgemeinen nicht, es sei denn, es liegen außergewöhnliche Umstände vor. Wenn du eine besondere Situation hast, kontaktiere das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support mit den Einzelheiten und der Begründung dafür und wir werden deine Anfrage prüfen.

Wird es neue Feature-Releases für geben? AWS OpsWorks for Puppet Enterprise

Nein. Da der Dienst das Ende seiner Nutzungsdauer erreicht, werden wir keine neuen Funktionen veröffentlichen. Wir werden jedoch weiterhin die Sicherheit verbessern und die Server bis zum Ende der Nutzungsdauer wie erwartet verwalten.

Erste Schritte mit OpsWorks für Puppet Enterprise

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

OpsWorks für Puppet Enterprise können Sie einen Puppet Enterprise-Server ausführen in. AWS Sie können in etwa 15 Minuten einen Puppet Enterprise Master-Server einrichten.

Ab dem 3. Mai 2021 speichert Puppet Enterprise einige Puppet Enterprise-Serverattribute in. OpsWorks AWS Secrets Manager Weitere Informationen finden Sie unter Integration in AWS Secrets Manager.

Die folgende exemplarische Vorgehensweise hilft Ihnen dabei, Ihren ersten Puppet-Master für Puppet Enterprise zu erstellen. OpsWorks

Voraussetzungen

Bevor Sie beginnen, müssen Sie die folgenden Voraussetzungen erfüllen.

Themen

Installieren des Puppet Development Kit

1. Laden Sie von der Puppet-Website das Puppet Development Kit für das Betriebssystem Ihres lokalen Computers herunter.
2. Installieren Sie das Puppet Development Kit.
3. Fügen Sie die Puppet Development Kit der PATH-Variablen Ihres lokalen Computers hinzu.

   - Auf einem Linux-oder macOS-Betriebssystem können Sie das Puppet Development Kit Ihrer PATH-Variablen hinzufügen, indem Sie den folgenden Befehl in einer Bash-Shell ausführen.

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

   - Auf einem Windows-basierten Betriebssystem können Sie das Puppet Development Kit zu Ihrer **PATH** Variablen hinzufügen, indem Sie den folgenden .NET Framework-Befehl in einer PowerShell Sitzung oder im Dialogfeld Umgebungsvariablen verwenden, auf das Sie über die Systemeigenschaften zugreifen können. Möglicherweise müssen Sie Ihre PowerShell Sitzung als Administrator ausführen, um den folgenden Befehl ausführen zu können.

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

Installieren der Puppet Enterprise Client-Tools

Puppet Enterprise (PE)-Client-Tools sind eine Sammlung von Befehlszeilen-Tools, mit denen Sie Zugriff auf Puppet Enterprise Services von Ihrer Workstation aus haben. Die Tools können auf vielen verschiedenen Betriebssystemen installiert werden, und sie können auch auf Knoten installiert werden, die Sie mit Puppet verwalten. Weitere Informationen zu den unterstützten Betriebssystemen für die Tools und wie sie installiert werden finden Sie unter Installing PE client tools in der Puppet Enterprise-Dokumentation.

Einrichten eines Git-Control-Repositorys

Bevor Sie einen Puppet-Master starten können, müssen Sie ein Steuerungs-Repository haben, das in Git konfiguriert wurde, um Ihre Puppet-Module und Klassen zu speichern und zu ändern/verwalten. In den Schritten zum Starten des Puppet-Enterprise-Master-Servers werden eine URL für ein Git-Repository und HTTPS- oder SSH-Kontoinformationen für den Zugriff auf das Repository benötigt. Weitere Informationen zum Einrichten eines Steuerungs-Repositorys, das Ihr Puppet Enterprise-Master verwendet, finden Sie unter Setting up a control repository. Anweisungen zur Einrichtung des Kontroll-Repositorys finden Sie auch in der Readme-Datei zum `control-repo`Beispiel-Repository von Puppet unter. GitHub Das Steuerungs-Repository ist etwa wie folgt aufgebaut.

```
### 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
```

Einrichten eines Repositorys mithilfe von CodeCommit

Sie können ein neues Repository erstellen, indem Sie CodeCommit Weitere Informationen CodeCommit zur Erstellung Ihres Kontroll-Repositorys finden Sie the section called “Optional: Verwenden CodeCommit” in dieser Anleitung. Weitere Informationen zu den ersten Schritten mit Git on CodeCommit finden Sie unter Erste Schritte mit AWS CodeCommit. Um Ihren Server OpsWorks für Puppet Enterprise für Ihr Repository zu autorisieren, fügen Sie die `AWSCodeCommitReadOnly` Richtlinie Ihrer IAM-Instanzprofilrolle hinzu.

Einrichten einer VPC

Ihr OpsWorks for Puppet Enterprise Master muss in einer Amazon Virtual Private Cloud betrieben werden. Sie können den Server zu einer vorhandenen VPC hinzufügen, die Standard-VPC verwenden oder eine neue VPC erstellen. Informationen zu Amazon VPC und zur Erstellung einer neuen VPC finden Sie im Amazon VPC Getting Started Guide.

Selbst erstellte oder vorhandene VPCs müssen folgende Einstellungen oder Eigenschaften aufweisen.

- Die VPC sollte über mindestens ein Subnetz verfügen.

  Wenn Ihr Master OpsWorks für Puppet Enterprise öffentlich zugänglich sein soll, machen Sie das Subnetz öffentlich und aktivieren Sie Automatische Zuweisung öffentlicher IP-Adressen.
- DNS resolution (DNS-Auflösung) muss aktiviert sein.
- Aktivieren Sie auf dem Subnetz Auto-assign public IP (Öffentliche IP automatisch zuweisen).

Wenn Sie nicht damit vertraut sind, VPCs zu erstellen oder Ihre Instances darin auszuführen, können Sie den folgenden AWS CLI Befehl ausführen, um eine VPC mit einem einzigen öffentlichen Subnetz zu erstellen, indem Sie eine AWS CloudFormation Vorlage verwenden, die AWS OpsWorks für Sie bereitgestellt wird. Wenn Sie lieber die verwenden möchten AWS Management Console, können Sie die Vorlage auch auf die Konsole hochladen. AWS CloudFormation

```
aws cloudformation create-stack --stack-name OpsWorksVPC --template-url https://
s3.amazonaws.com/opsworks-cm-us-east-1-prod-default-assets/misc/opsworks-cm-vpc.yaml
```

Einrichten eines EC2-Schlüsselpaares (optional)

Eine SSH-Verbindung ist für die typische Verwaltung des Puppet-Servers nicht erforderlich oder wird nicht empfohlen. Sie können die AWS CLI Befehle AWS Management Console und verwenden, um viele Verwaltungsaufgaben auf Ihrem Puppet-Server auszuführen.

Ein EC2-Schlüsselpaar ist erforderlich, um eine SSH-Verbindung mit dem Server herzustellen, wenn Sie das Passwort zum Anmelden bei der web-basierten Puppet Enterprise-Konsole vergessen haben oder ändern möchten. Sie können ein bestehendes Schlüsselpaar verwenden oder ein neues Schlüsselpaar erstellen. Weitere Informationen zum Erstellen eines neuen EC2-Schlüsselpaars finden Sie unter Amazon EC2 EC2-Schlüsselpaare.

Wenn Sie kein EC2-Schlüsselpaar benötigen, sind Sie bereit, einen Puppet Enterprise-Master zu erstellen.

Voraussetzungen für die Verwendung einer benutzerdefinierten Domäne (optional)

Sie können Ihren Puppet Enterprise-Master in Ihrer eigenen Domäne einrichten und einen öffentlichen Endpunkt in einer benutzerdefinierten Domäne angeben, der als Endpunkt Ihres Servers verwendet werden soll. Wenn Sie eine benutzerdefinierte Domäne verwenden, sind alle der folgenden Anforderungen erforderlich, wie in diesem Abschnitt ausführlich beschrieben.

Themen

- Einrichten einer benutzerdefinierten Domäne
- Anfordern eines Zertifikats
- Anfordern eines privaten Schlüssels

Einrichten einer benutzerdefinierten Domäne

Um Ihren Puppet Enterprise-Master in Ihrer eigenen benutzerdefinierten Domäne ausführen zu können, benötigen Sie einen öffentlichen Endpunkt eines Servers, z. B. `https://aws.my-company.com`. Wenn Sie eine benutzerdefinierte Domäne angeben, müssen Sie auch ein Zertifikat und einen privaten Schlüssel bereitstellen, wie in den vorherigen Abschnitten beschrieben.

Um auf den Server zuzugreifen, nachdem Sie ihn erstellt haben, fügen Sie einen CNAME-DNS-Eintrag in Ihrem bevorzugten DNS-Dienst hinzu. Dieser Datensatz muss die benutzerdefinierte Domäne auf den Endpunkt (den Wert des Serverattributs `Endpoint`) verweisen, der durch den Erstellungsprozess des Puppet-Masters generiert wird. Sie können nicht mit dem generierten

`Endpoint`-Wert auf den Server zugreifen, wenn der Server eine benutzerdefinierte Domain verwendet.

Anfordern eines Zertifikats

Zum Einrichten Ihres Puppet-Masters in Ihrer eigenen benutzerdefinierten Domäne benötigen Sie ein PEM-formatiertes HTTPS-Zertifikat. Dabei kann es sich um ein einzelnes, selbstsigniertes Zertifikat oder um eine Zertifikatkette handeln. Wenn Sie dieses Zertifikat angeben, müssen Sie auch eine benutzerdefinierte Domäne und einen privaten Schlüssel angeben, wenn Sie den Workflow Create a Puppet Enterprise Master (Puppet Enterprise-Master erstellen) durchführen.

Für den Zertifikatwert gelten folgende Anforderungen:

- Sie können entweder ein selbstsigniertes, benutzerdefiniertes Zertifikat oder die vollständige Zertifikatkette bereitstellen.
- Das Zertifikat muss ein gültiges X509-Zertifikat oder eine Zertifikatkette im PEM-Format sein.
- Das Zertifikat muss zum Zeitpunkt des Hochladens gültig sein. Sie können ein Zertifikat nicht vor Beginn des Gültigkeitszeitraums (das Datum `NotBefore` des Zertifikats) oder nach Ablauf der Gültigkeit (das Datum `NotAfter` des Zertifikats) hochladen.
- Der allgemeine Name des Zertifikats oder die alternativen Antragstellernamen (SANs) des Zertifikats müssen, sofern vorhanden, mit dem benutzerdefinierten Domänenwert übereinstimmen.
- Das Zertifikat muss mit dem Wert des Felds Custom private key (Benutzerdefinierter privater Schlüssel) übereinstimmen.

Anfordern eines privaten Schlüssels

Um Ihren Puppet-Master in Ihrer eigenen benutzerdefinierten Domäne einzurichten, benötigen Sie einen privaten Schlüssel im PEM-Format, um eine Verbindung mit dem Server mithilfe von HTTPS herzustellen. Der private Schlüssel darf nicht verschlüsselt sein; er kann nicht durch ein Passwort oder eine Passphrase geschützt werden. Wenn Sie einen benutzerdefinierten privaten Schlüssel angeben, müssen Sie auch eine benutzerdefinierte Domäne und ein Zertifikat bereitstellen.

Erstellen eines Puppet Enterprise-Masters

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden

deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

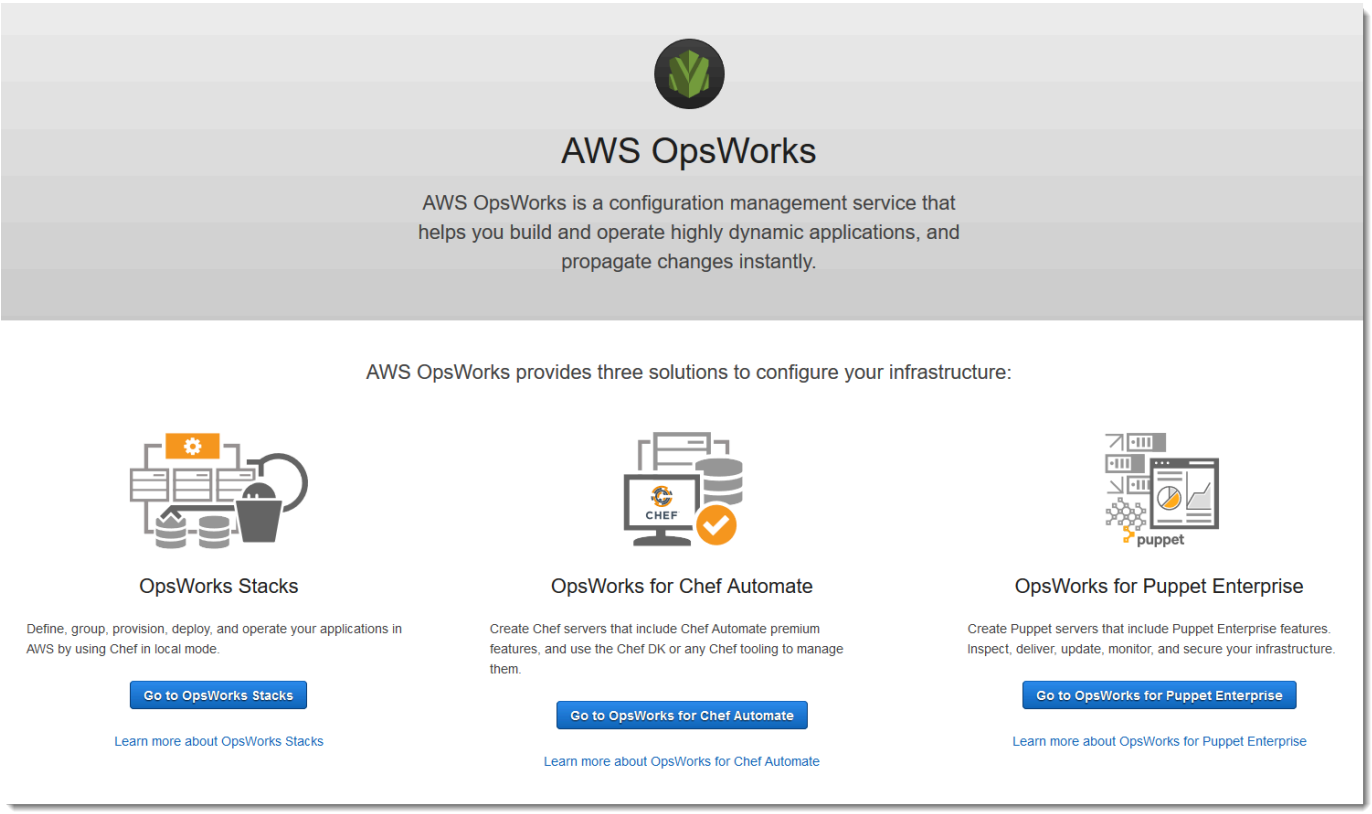
Sie können einen Puppet-Master mithilfe der OpsWorks for Puppet Enterprise-Konsole oder der erstellen. AWS CLI

Themen

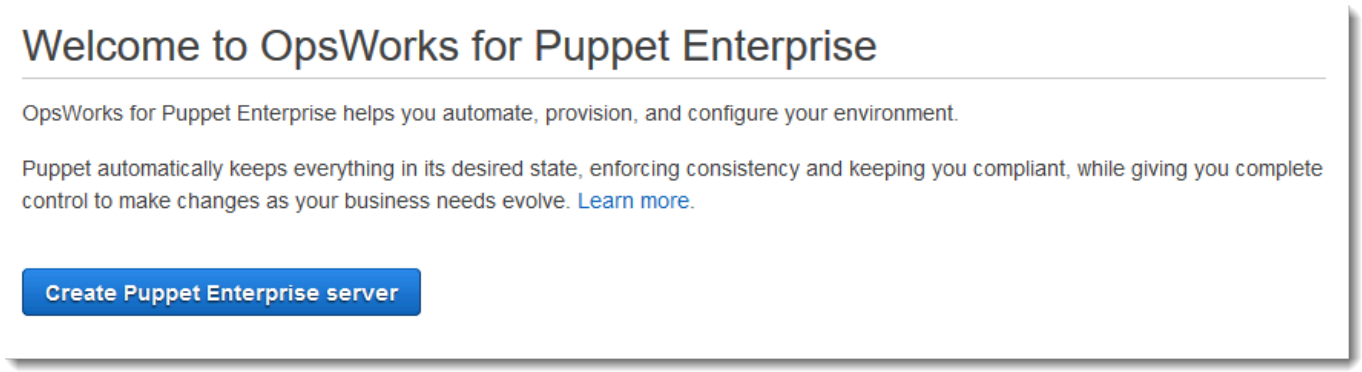
- Erstellen Sie einen Puppet Enterprise Master mithilfe der AWS Management Console
- Erstellen Sie einen Puppet Enterprise Master mit dem AWS CLI

Erstellen Sie einen Puppet Enterprise Master mithilfe der AWS Management Console

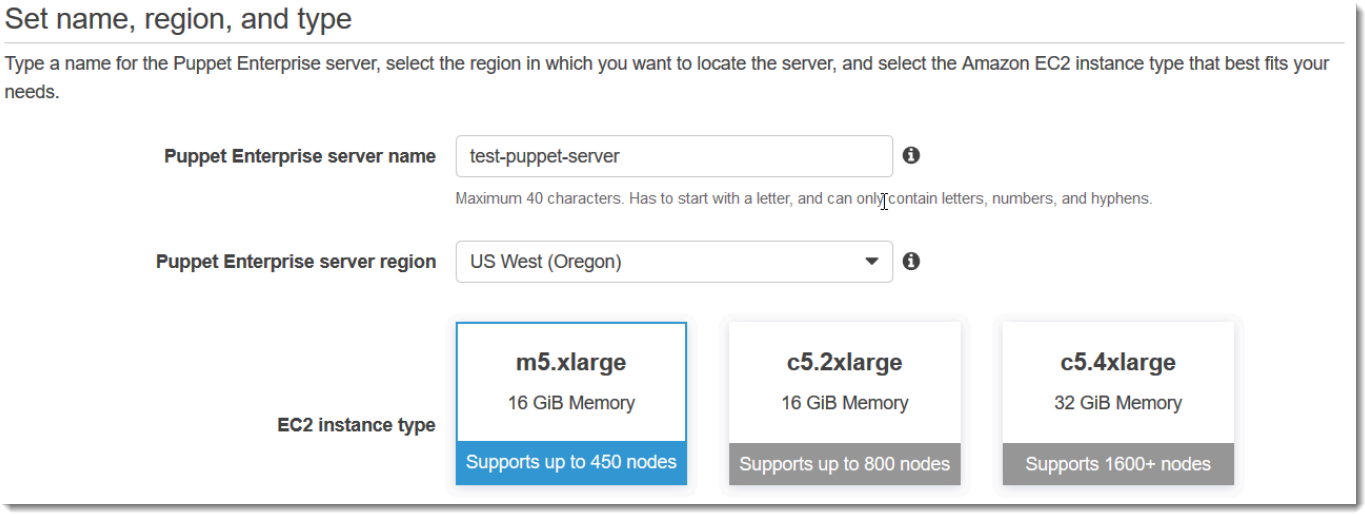
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS OpsWorks Konsole unter https://console.aws.amazon.com/opsworks/.
2. Wählen Sie auf der AWS OpsWorks Startseite Go to OpsWorks for Puppet Enterprise aus.



3. Wählen Sie auf der OpsWorks Startseite von Puppet Enterprise die Option Create Puppet Enterprise Server aus.

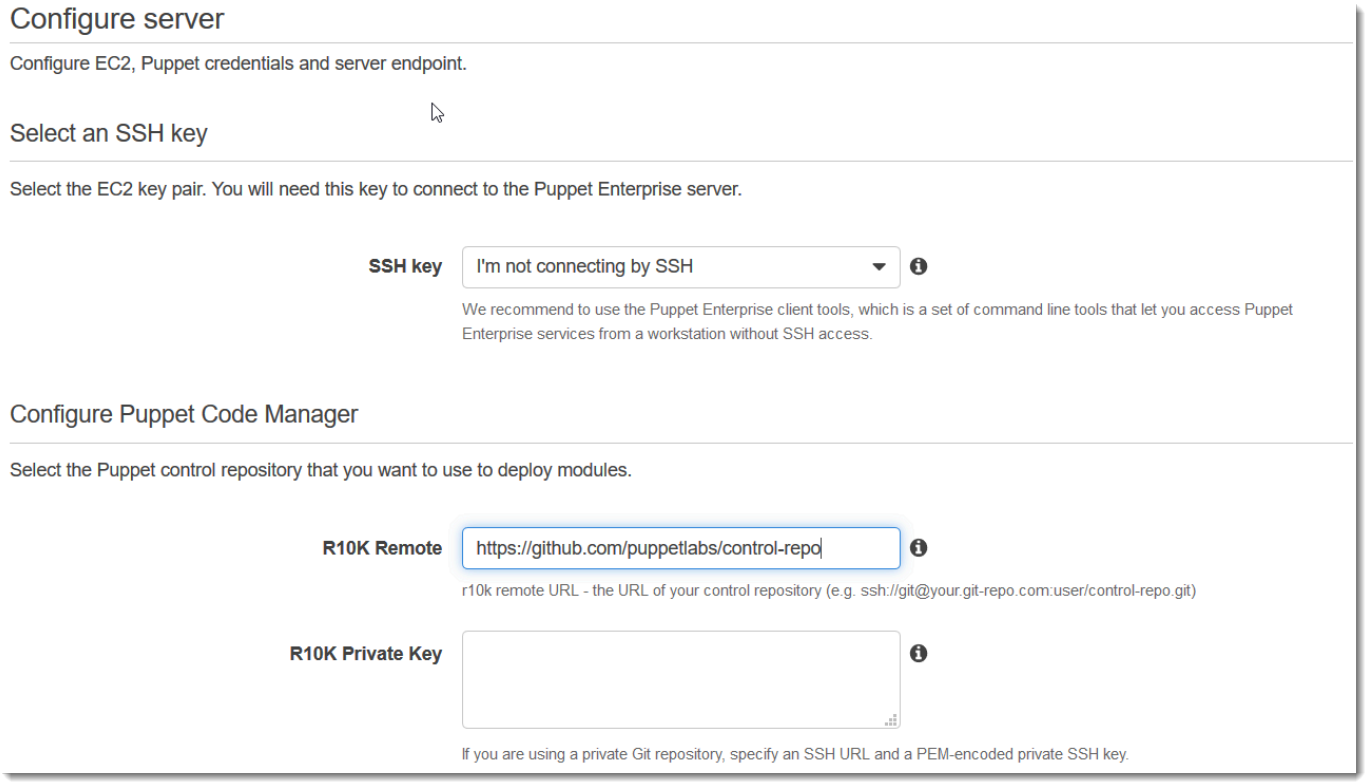


4. Geben Sie auf der Seite Set name, region, and type (Name, Region und Typ festlegen) einen Namen für Ihren Server an. Puppet Master-Namen dürfen maximal 40 Zeichen lang sein, müssen mit einem Buchstaben beginnen und dürfen nur alphanumerische Zeichen und Bindestriche enthalten. Wählen Sie erst eine unterstützte Region und dann einen Instance-Typ aus, der die Anzahl der Knoten unterstützt, die Sie verwalten möchten. Sie können bei Bedarf den Instance-Typ ändern, nachdem Ihr Server erstellt wurde. Für diese exemplarische Vorgehensweise erstellen wir einen Instance-Typ m5.xlarge in der Region USA West (Oregon). Wählen Sie Weiter aus.



5. Behalten Sie auf der Seite Configure server (Server konfigurieren) in der Dropdown-Liste SSH key (SSH-Schlüssel) die Standardeinstellung bei, es sei denn, Sie möchten ein Schlüsselpaar angeben. Geben Sie im Feld r10k remote des Bereichs Configure Puppet Code Manager eine gültige SSH- oder HTTPS-URL Ihres Git Remote-Repositorys an. Fügen Sie im Feld für den privaten R10K-Schlüssel den privaten SSH-Schlüssel ein, mit dem Sie auf das R10k-Remote-Repository AWS OpsWorks zugreifen können. Dieser wird von Git bereitgestellt, wenn Sie ein

privates Repository erstellen, ist aber nicht erforderlich, wenn Sie HTTPS-Authentifizierung für den Zugriff auf Ihr Steuerungs-Repository verwenden. Wählen Sie Weiter aus.



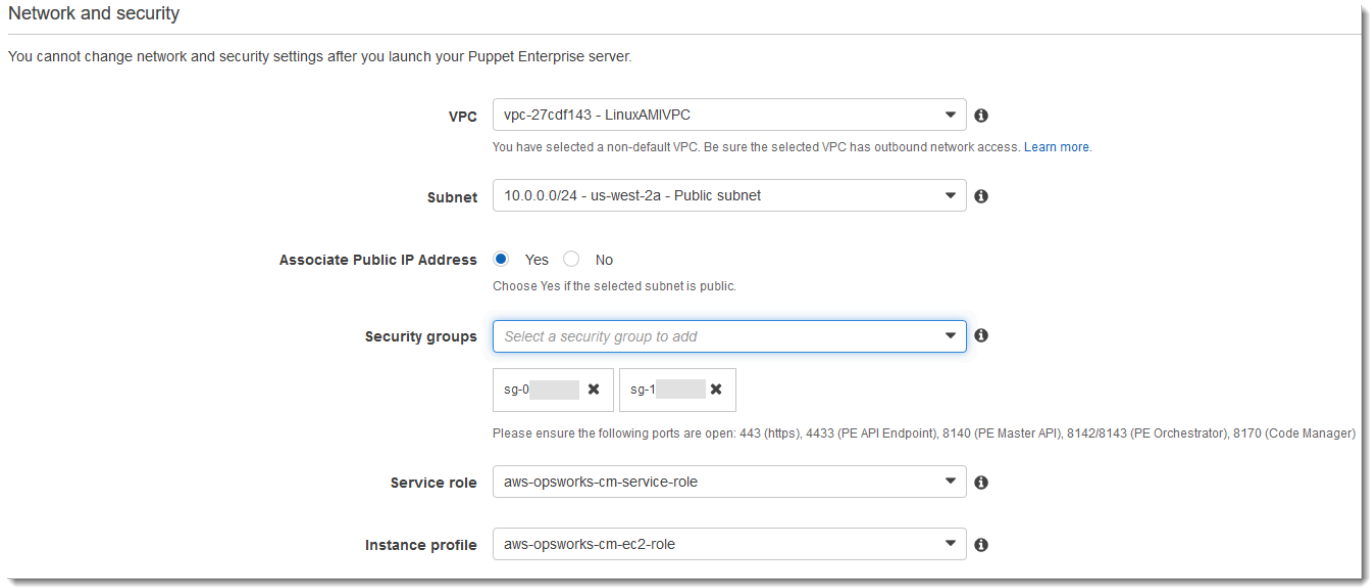
6. Behalten Sie unter Specify server endpoint (Serverendpunkt angeben) die Standardeinstellung Use an automatically generated endpoint (Automatisch generierten Endpunkt verwenden) bei und wählen Sie dann Next (Weiter), es sei denn, der Server soll sich in einer eigenen benutzerdefinierten Domäne befinden. Fahren Sie mit dem nächsten Schritt fort, um eine benutzerdefinierte Domäne zu konfigurieren.
7. Um eine benutzerdefinierte Domäne zu verwenden, wählen Sie unter Specify server endpoint (Serverendpunkt angeben) aus der Dropdown-Liste die Option Use a custom domain (Benutzerdefinierte Domäne verwenden) aus.

   a. Geben Sie für Fully Qualified Domain Name (FQDN) einen FQDN an. Sie müssen Eigentümer des Domänennamens sein, den Sie verwenden möchten.

   b. Fügen Sie unter SSL certificate (SSL-Zertifikat) das gesamte PEM-formatierte Zertifikat ein, beginnend mit `-----BEGIN CERTIFICATE-----` und endend mit `-----END CERTIFICATE-----`. Der Antragssteller des SSL-Zertifikats muss mit dem FQDN übereinstimmen, den Sie im vorherigen Schritt eingegeben haben. Entfernen Sie alle zusätzlichen Zeilen vor und hinter dem Zertifikat.

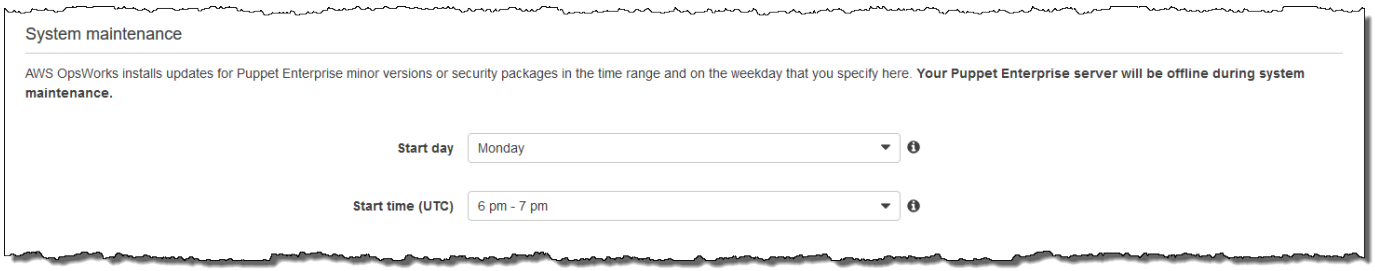
c. Fügen Sie unter SSL private key (Privater SSL-Schlüssel) den gesamten privaten RSA-Schlüssel ein, beginnend mit `-----BEGIN RSA PRIVATE KEY-----` und endend mit `-----END RSA PRIVATE KEY-----`. Der private SSL-Schlüssel muss mit dem öffentlichen Schlüssel im SSL-Zertifikat übereinstimmen, das Sie im vorherigen Schritt eingegeben haben. Entfernen Sie alle zusätzlichen Zeilen vor und hinter dem privaten Schlüssel. Wählen Sie Weiter aus.

8. Wählen Sie auf der Seite Erweiterte Einstellungen konfigurieren im Bereich Netzwerk und Sicherheit eine VPC, ein Subnetz und eine oder mehrere Sicherheitsgruppen aus. AWS OpsWorks kann eine Sicherheitsgruppe, eine Servicerolle und ein Instanzprofil für Sie generieren, falls Sie noch keine haben, die Sie verwenden möchten. Der Server kann zu mehreren Sicherheitsgruppen gehören. Sie können die Netzwerk- und Sicherheitseinstellungen für den Puppet-Master nicht ändern, nachdem Sie diese Seite verlassen haben.

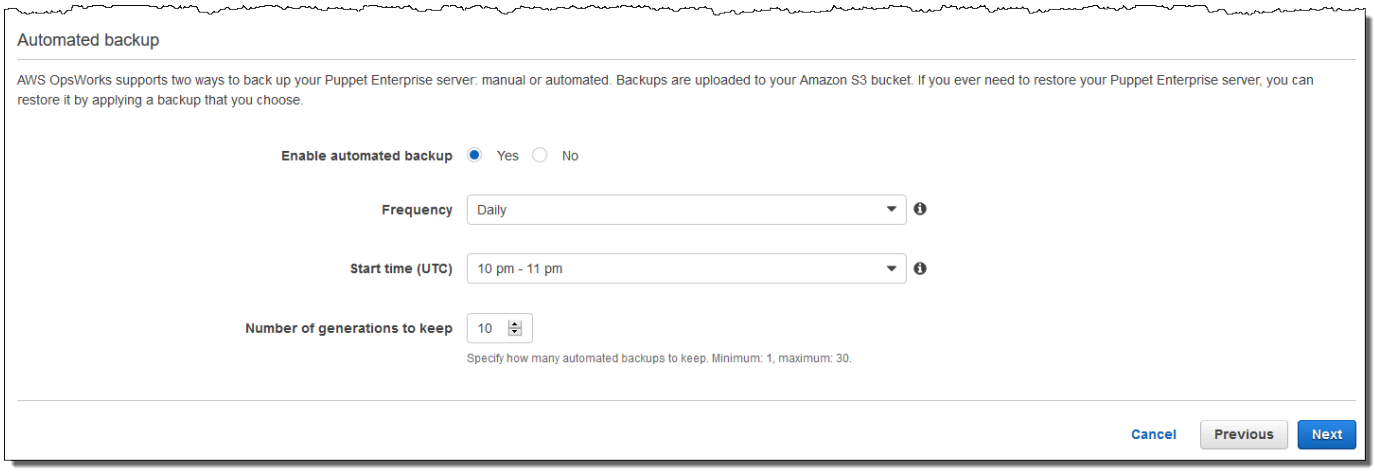


9. Legen Sie im Abschnitt System maintenance (Systemwartung) den Tag und die Uhrzeit fest, zu der die Systemwartung beginnen soll. Da der Server während der Systemwartung offline ist, wählen Sie eine Uhrzeit innerhalb der normalen Geschäftszeiten mit geringer Server-Nachfrage aus.

   Das Wartungsfenster muss angegeben werden. Sie können den Starttag und die Startzeit später ändern AWS Management Console, indem Sie die APIs AWS CLI, oder verwenden.



10. Konfigurieren Sie die Sicherungen. Standardmäßig sind automatische Sicherungen aktiviert. Legen Sie eine bevorzugte Häufigkeit und Stunde für den Start der automatischen Sicherung sowie die Anzahl der Backup-Generationen fest, die in Amazon Simple Storage Service gespeichert werden sollen. Es können maximal 30 Backups aufbewahrt werden. Wenn das Maximum erreicht ist, OpsWorks löscht Puppet Enterprise die ältesten Backups, um Platz für neue zu schaffen.



11. (Optional) Fügen Sie dem Server und den verwandten Ressourcen unter Tags Tags hinzu, z. B. die EC2-Instance, die Elastic IP-Adresse, die Sicherheitsgruppe, den S3-Bucket und Sicherungen. Weitere Informationen zum Markieren eines Puppet OpsWorks Enterprise-Servers finden Sie unter. Mit Tags auf AWS OpsWorks for Puppet Enterprise Ressourcen arbeiten
12. Wenn Sie die erweiterten Einstellungen fertig konfiguriert haben, wählen Sie Next (Weiter) aus.
13. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Wenn Sie bereit sind, den Server zu erstellen, wählen Sie Launch (Starten) aus.

    Während Sie darauf warten, Ihren Puppet Master AWS OpsWorks zu erstellen, fahren Sie mit dem Starter Kit Konfigurieren des Puppet-Masters mit dem Starter Kit und den Zugangsdaten für die Puppet Enterprise-Konsole fort und laden Sie sie herunter. Warten Sie nicht mit dem Herunterladen dieser Elemente, bis Ihre Server online ist.

Wenn die Servererstellung abgeschlossen ist, ist Ihr Puppet-Master auf der OpsWorks Startseite von Puppet Enterprise mit dem Status Online verfügbar. Nachdem sich der Server online befindet, ist die Puppet Enterprise-Konsole auf der Server-Domäne mit einer URL mit folgendem Format verfügbar: `https://your_server_name-randomID.region.opsworks-cm.io`.

Erstellen Sie einen Puppet Enterprise Master mit dem AWS CLI

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Das Erstellen eines Masterservers OpsWorks für Puppet Enterprise durch Ausführen von AWS CLI Befehlen unterscheidet sich vom Erstellen eines Servers in der Konsole. AWS OpsWorks Erstellt in der Konsole eine Servicerolle und eine Sicherheitsgruppe für Sie, falls Sie keine vorhandenen angeben, die Sie verwenden möchten. In der AWS OpsWorks kann eine Sicherheitsgruppe für Sie erstellen AWS CLI, wenn Sie keine angeben, aber es wird nicht automatisch eine Dienstrolle erstellt. Sie müssen einen Dienstrollen-ARN als Teil Ihres `create-server` Befehls angeben. In der Konsole laden Sie während AWS OpsWorks der Erstellung Ihres Puppet-Masters das Starterkit und die Anmeldeinformationen für die Puppet Enterprise-Konsole herunter. Da Sie dies nicht tun können, wenn Sie einen Master OpsWorks für Puppet Enterprise mithilfe von erstellen AWS CLI, verwenden Sie ein JSON-Verarbeitungsprogramm, um die Anmeldeinformationen und das Starterkit aus den Ergebnissen des `create-server` Befehls abzurufen, nachdem Ihr neuer OpsWorks für Puppet Enterprise-Master online ist.

Wenn auf Ihrem lokalen Computer das noch nicht ausgeführt wird AWS CLI, laden Sie es herunter und installieren Sie es, AWS CLI indem Sie den Installationsanweisungen im AWS-Benutzerhandbuch für die Befehlszeilenschnittstelle folgen. In diesem Abschnitt werden nicht alle Parameter beschrieben, die Sie mit dem Befehl `create-server` verwenden können. Weitere Informationen zu den `create-server`-Parametern finden Sie unter `create-server` in der AWS CLI -Referenz.

1. Schließen Sie die Voraussetzungen ab. Zum Erstellen eines Puppet-Masters benötigen Sie eine Subnetz-ID und somit eine VPC.
2. Erstellen Sie eine Servicerolle und ein Instanzprofil. AWS OpsWorks stellt eine AWS CloudFormation Vorlage bereit, mit der Sie beide erstellen können. Führen Sie den folgenden AWS CLI Befehl aus, um einen AWS CloudFormation Stack zu erstellen, der die Servicerolle und das Instanzprofil für Sie erstellt.

```
aws cloudformation create-stack --stack-name OpsWorksCMRoles --template-url
 https://s3.amazonaws.com/opsworks-cm-us-east-1-prod-default-assets/misc/opsworks-
cm-roles.yaml --capabilities CAPABILITY_NAMED_IAM
```

3. Suchen Sie nach AWS CloudFormation Abschluss der Erstellung des Stacks die ARNs der Servicerollen in Ihrem Konto und kopieren Sie sie.

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

Suchen Sie in den Ergebnissen des Befehls `list-roles` nach Servicerollen-ARN-Einträgen, die dem folgenden ähneln. Notieren Sie sich die Servicerollen-ARNs. Sie benötigen diese Werte zum Erstellen Ihres Puppet Enterprise-Masters.

```
{
    "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"
}
```

4. Suchen und kopieren Sie die ARNs von Instance-Profilen in Ihrem Konto.

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

Suchen Sie in den Ergebnissen des Befehls `list-instance-profiles` nach Instance-Profil-ARN-Einträgen, die dem folgenden ähneln. Notieren Sie sich die ARNs der Instance-Profile. Sie benötigen diese Werte zum Erstellen Ihres Puppet Enterprise-Masters.

```
{
    "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. Erstellen Sie den Master OpsWorks für Puppet Enterprise, indem Sie den `create-server` Befehl ausführen.

   - Der `--engine` Wert ist `PuppetMonolithic`, `--engine-model` ist und `--engine-version` kann `2019` oder `2017` sein.
   - Der Servername muss innerhalb Ihres AWS Kontos in jeder Region eindeutig sein. Servernamen müssen mit einem Buchstaben beginnen. Danach können Buchstaben, Zahlen und Bindestriche (-) verwendet werden, insgesamt höchstens 40 Zeichen.
   - Verwenden Sie die ARNs des Instance-Profils und der Servicerolle, die Sie in Schritt 3 und 4 kopiert haben.
   - Gültige Instance-Typen sind `m5.xlarge`, `c5.2xlarge` und `c5.4xlarge`. Weitere Informationen zu den Spezifikationen dieser Instance-Typen finden Sie unter Instance-Typen im Amazon EC2 EC2-Benutzerhandbuch.
   - Der Parameter `--engine-attributes` ist optional. Wenn Sie kein Puppet-Administratorpasswort festlegen, wird bei der Servererstellung ein Passwort generiert. Wenn Sie `--engine-attributes` hinzufügen, geben Sie für `PUPPET_ADMIN_PASSWORD` ein Administratorpasswort für die Anmeldung auf der Puppet Enterprise-Konsolenwebseite ein. Das Passwort muss zwischen 8 und 32 ASCII-Zeichen lang sein.
   - Ein SSH-Schlüsselpaar ist optional. Es kann Ihnen dabei helfen, sich mit dem Puppet-Master zu verbinden, wenn Sie das Konsolenadministratorpasswort zurücksetzen müssen. Weitere Informationen zum Erstellen eines SSH-Schlüsselpaars finden Sie unter Amazon EC2 EC2-Schlüsselpaare im Amazon EC2 EC2-Benutzerhandbuch.

- Um eine benutzerdefinierte Domäne zu verwenden, fügen Sie dem Befehl die folgenden Parameter hinzu. Andernfalls generiert der Erstellungsprozess des Puppet-Master automatisch einen Endpunkt für Sie. Alle drei Parameter sind erforderlich, um eine benutzerdefinierte Domäne zu konfigurieren. Informationen zu zusätzlichen Anforderungen für die Verwendung dieser Parameter finden Sie CreateServerin der AWS OpsWorks CM-API-Referenz.
  - `--custom-domain` – Ein optionaler öffentlicher Endpunkt eines Servers, z. B. `https://aws.my-company.com`.
  - `--custom-certificate` – Ein PEM-formatiertes HTTPS-Zertifikat. Der Wert kann ein einzelnes, selbstsigniertes Zertifikat oder eine Zertifikatkette sein.
  - `--custom-private-key` – Ein privater Schlüssel im PEM-Format für die Verbindung mit dem Server mithilfe von HTTPS. Der private Schlüssel darf nicht verschlüsselt sein; er kann nicht durch ein Passwort oder eine Passphrase geschützt werden.
- Es ist eine wöchentliche Systemwartung erforderlich. Gültige Werte müssen im folgenden Format angegeben werden: `DDD:HH:MM`. Die angegebene Uhrzeit entspricht der Zeitzone UTC (Coordinated Universal Time). Wenn Sie für `--preferred-maintenance-window` keinen Wert angeben, wird ein zufälliger Standardwert mit einem einstündigen Zeitraum an einem Dienstag, Mittwoch oder Freitag festgelegt.
- Gültige Werte für `--preferred-backup-window` müssen in einem der folgenden Formate angegeben werden: `HH:MM` für tägliche Sicherungen oder `DDD:HH:MM` für wöchentliche Sicherungen. Die angegebene Uhrzeit entspricht der Zeitzone UTC. Standardmäßig wird ein zufälliger täglicher Startzeitpunkt festgelegt. Wenn Sie automatische Sicherungen deaktivieren möchten, verwenden Sie stattdessen den Parameter `--disable-automated-backup`.
- Geben Sie für `--security-group-ids` eine oder mehrere Sicherheitsgruppen-IDs, durch Kommata getrennt, ein.
- Geben Sie für `--subnet-ids` eine Subnetz-ID ein.

```
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
```

Im Folgenden wird ein Beispiel gezeigt.

```
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
```

Im folgenden Beispiel wird ein Puppet-Master erstellt, der eine benutzerdefinierte Domäne verwendet.

```
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
```

Im folgenden Beispiel wird ein Puppet-Master erstellt, der zwei Tags hinzufügt: `Stage: Production` und `Department: Marketing`. Weitere Informationen zum Hinzufügen und Verwalten von Tags OpsWorks für Puppet Enterprise-Server finden Sie Mit Tags auf AWS OpsWorks for Puppet Enterprise Ressourcen arbeiten in diesem Handbuch.

```
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 für Puppet Enterprise dauert die Erstellung eines neuen Servers etwa 15 Minuten. Sie sollten die Ausgabe des Befehls `create-server` nicht verwerfen oder die Shell-Sitzung beenden, da die Ausgabe wichtige Informationen enthalten kann, die nicht wiederhergestellt werden können. Um Passwörter und das Starter Kit aus den Ergebnissen von `create-server` zu extrahieren, fahren Sie mit dem nächsten Schritt fort.

   Wenn Sie eine benutzerdefinierte Domäne mit dem Server verwenden, kopieren Sie in der Ausgabe des Befehls `create-server` den Wert des Attributs `Endpoint`. Im Folgenden wird ein Beispiel gezeigt.

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

7. Wenn Sie sich dafür entschieden haben OpsWorks , dass Puppet Enterprise ein Passwort für Sie generiert, können Sie es mithilfe eines JSON-Prozessors wie jq in einem verwendbaren Format aus den `create-server` Ergebnissen extrahieren. Nachdem Sie jq installiert haben, können Sie die folgenden Befehle ausführen, um das Puppet-Administratorpasswort und das Starter Kit zu extrahieren. Wenn Sie in Schritt 3 kein eigenes Passwort angegeben haben, speichern Sie das extrahierte Administratorpasswort an einem sicheren Speicherort.

   ```
   #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**
>
> Es ist nicht möglich, in der AWS Management Console ein neues Starter Kit für den Puppet-Master zu erstellen. Wenn Sie mit dem einen Puppet-Master erstellen AWS CLI, führen Sie den vorherigen `jq` Befehl aus, um das Base64-kodierte Starterkit in den `create-server` Ergebnissen als ZIP-Datei zu speichern.

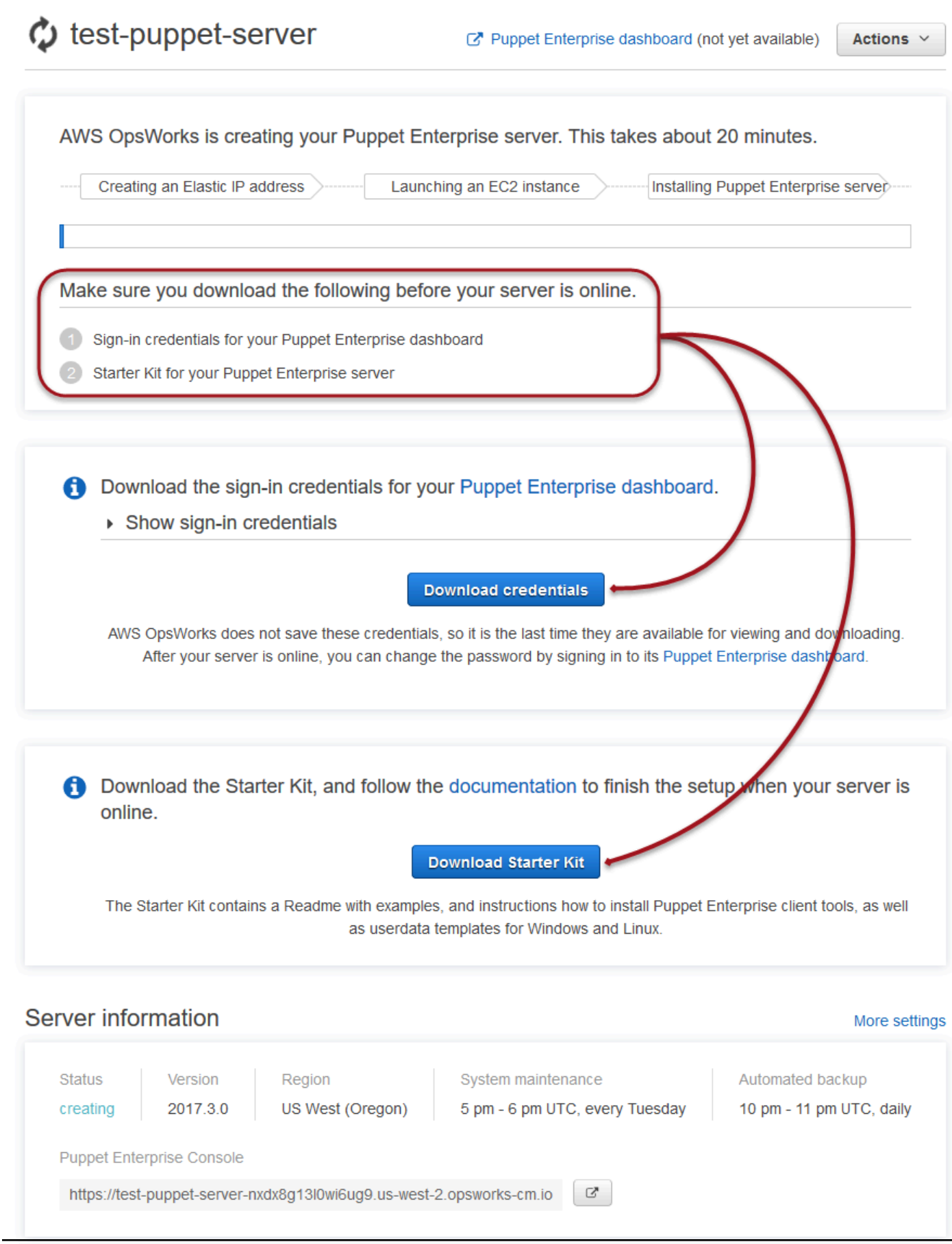
8. Wenn Sie keine benutzerdefinierte Domäne verwenden, fahren Sie mit dem nächsten Schritt fort. Wenn Sie eine benutzerdefinierte Domäne mit dem Server verwenden, erstellen Sie einen CNAME-Eintrag im DNS-Verwaltungstool Ihres Unternehmens, um Ihre benutzerdefinierte Domäne auf den OpsWorks for Puppet Enterprise-Endpunkt zu verweisen, den Sie in Schritt 6 kopiert haben. Sie können einen Server erst dann mit einer benutzerdefinierten Domäne erreichen und sich erst dann bei ihm anmelden, nachdem Sie diesen Schritt ausgeführt haben.
9. Fahren Sie mit dem nächsten Abschnitt, the section called “Beenden der Konfiguration” fort.

Konfigurieren des Puppet-Masters mit dem Starter Kit

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Während die Erstellung des Puppet-Masters noch läuft, wird die Eigenschaftenseite des Servers in der OpsWorks for Puppet Enterprise-Konsole geöffnet. Wenn Sie zum ersten Mal mit einem neuen Puppet-Master arbeiten, werden Sie von der Eigenschaftenseite aufgefordert, die erforderlichen

Elemente herunterzuladen. Laden Sie diese Elemente herunter, bevor Ihr Puppet-Server online geht. Die Schaltflächen zum Herunterladen sind nicht verfügbar, wenn ein neuer Server online ist.


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

- Anmeldeinformationen für den Puppet-Master. Sie werden diese Anmeldeinformationen verwenden, um sich bei der Puppet Enterprise-Konsole anzumelden, wo Sie die meisten Knotenverwaltungen durchführen. AWS OpsWorks speichert diese Anmeldeinformationen nicht. Dies ist das letzte Mal, dass sie angezeigt und heruntergeladen werden können. Falls erforderlich, können Sie das Passwort ändern, nachdem Sie sich mit diesen Anmeldeinformationen angemeldet haben.
- Starter Kit. Das Starter Kit enthält eine Readme-Datei mit Informationen und Beispielen, wo beschrieben wird, wie die Einrichtung und die Administratoranmeldeinformationen für die Puppet Enterprise-Konsole fertiggestellt werden können. Jedes Mal, wenn Sie das Starter Kit herunterladen, werden neue Anmeldeinformationen erstellt, und die alten Anmeldeinformationen werden ungültig gemacht.

Voraussetzungen

1. Laden Sie während der Servererstellung die Anmeldeinformationen für den Puppet-Master herunter und speichern Sie diese an einem sicheren, aber leicht zugreifbaren Ort.
2. Laden Sie das Starter Kit herunter und extrahieren Sie die .zip-Datei des Starter Kits in Ihr Workspace-Verzeichnis. Geben Sie Ihre Anmeldeinformationen an niemanden weiter. Wenn andere Benutzer den Puppet-Master verwalten, fügen Sie diese zu einem späteren Zeitpunkt als Administratoren in der Puppet Enterprise-Konsole hinzu. Weitere Informationen zum Hinzufügen der Puppet-Master-Benutzer finden Sie unter Creating and managing users and user roles in der Puppet Enterprise-Dokumentation.

Installieren des Puppet-Master-Zertifikats

Um mit Ihrem Puppet-Master arbeiten und Knoten zur Verwaltung hinzufügen zu können, müssen Sie sein Zertifikat installieren. Installieren Sie es, indem Sie den folgenden AWS CLI Befehl ausführen. Sie können diese Aufgabe nicht in der ausführen 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
```

Erstellen eines Kurzzeit-Token

Zur Verwendung der Puppet-API müssen Sie einen Kurzzeit-Token für sich selbst erstellen. Dieser Schritt ist für die Verwendung der Puppet Enterprise-Konsole nicht erforderlich. Erstellen Sie das Token mit dem folgenden Befehl.

Die Standard-Nutzungsdauer für ein Token beträgt fünf Minuten. Sie können diese Standardeinstellung jedoch ändern.

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

> **Note**
>
> Da die Standard-Nutzungsdauer für Token fünf Minuten beträgt, fügt der vorherige Beispiel-Befehl den `--lifetime`-Parameter hinzu, um die Nutzungsdauer des Token zu verlängern. Sie können die Nutzungsdauer für das Token auf einen Zeitraum von bis zu 10 Jahren setzen (`10y`). Weitere Informationen darüber, wie Sie die Standard-Nutzungsdauer für ein Token ändern, finden Sie unter Change the token's default lifetime in der Puppet Enterprise-Dokumentation.

Richten Sie das Starter Kit ein (Apache-Beispiel)

Nachdem Sie das Starter Kit heruntergeladen und entpackt haben, können Sie den Beispielzweig im mitgelieferten `control-repo-example` Beispielordner verwenden, um einen Apache-Webserver auf Ihren verwalteten Knoten zu konfigurieren.

Das Starter Kit umfasst zwei `control-repo` Ordner: `control-repo` und `control-repo-example`. Der `control-repo` Ordner enthält einen `production` Zweig, der gegenüber dem, was Sie im GitHub Puppet-Repository sehen würden, unverändert ist. Der `control-repo-example` Ordner hat auch einen `production` Zweig, der Beispielcode zum Einrichten eines Apache-Servers mit einer Testwebsite enthält.

1. Verschieben Sie den `control-repo-example` `production`-Zweig in Ihr Git Remote-Repository (die `r10k_remote`-URL Ihres Puppet-Masters). Führen Sie in Ihrem Starter Kit-Stammverzeichnis den folgenden Befehl aus und ersetzen Sie *`dabei r10 kRemoteUrl`* durch Ihre `r10k_remote` URL.

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

Der Puppet-Code-Manager verwendet Git-Zweige als Umgebungen. Standardmäßig befinden sich alle Knoten in der Produktionsumgebung.

> **Important**
>
> Verschieben Sie ihn nicht in einen `master`-Zweig. Der `master`-Zweig ist für den Puppet-Master reserviert.

2. Stellen Sie den Code im `control-repo-example`-Zweig Ihrem Puppet Master bereit. Auf diese Weise kann der Puppet-Master Ihren Puppet-Code von Ihrem Git-Repository (`r10k_remote`) herunterladen. In Ihrem Starter Kit Root-Verzeichnis führen Sie Folgendes aus.

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

Weitere Informationen darüber, wie Sie die Apache-Beispielkonfiguration auf verwaltete Knoten anwenden können, die Sie in Amazon EC2 erstellen, finden Sie Schritt 2: Erstellen von Instances mit einem unbeaufsichtigten Skript für die Zuordnung in diesem Handbuch.

Hinzufügen von Knoten, die vom Puppet-Master verwaltet werden

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Service hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Themen

- Weitere Informationen

Die empfohlene Methode zum Hinzufügen von Knoten ist die Verwendung der AWS OpsWorks `associateNode()` API. Der Puppet Enterprise-Master-Server hostet ein Repository, das Sie verwenden, um die Puppet Agent-Software auf Knoten zu installieren, die Sie verwalten möchten. Dabei spielt es keine Rolle, ob sich die Knoten auf lokalen Computern oder virtuellen Maschinen befinden. Die Puppet-Agent-Software für einige Betriebssysteme wird im Rahmen des Startvorgangs auf dem OpsWorks for Puppet Enterprise-Server installiert. In der folgenden Tabelle sind die Betriebssystem-Agenten aufgeführt, die beim Start auf Ihrem Server OpsWorks für Puppet Enterprise verfügbar sind.

Vorinstallierte Betriebssystem-Agenten

Unterstütztes Betriebssystem	Versionen
Ubuntu	16.04, 18.04, 20.04
Red Hat Enterprise Linux (RHEL)	6, 7, 8
Windows	64-Bit-Editionen aller von Puppet unterstützten Windows-Versionen

Sie können `puppet-agent` Ihrem Server für andere Betriebssysteme hinzufügen. Beachten Sie, dass die Systemwartung Agenten löscht, die Sie Ihrem Server nach dem Start hinzugefügt haben. Obwohl die meisten vorhandenen zugeordneten Knoten, auf denen der gelöschte Agent bereits läuft, weiterhin angemeldet werden, kann es sein, dass Knoten mit Debian-Betriebssystemen keine Berichte mehr erstellen. Es wird empfohlen, die Installation manuell `puppet-agent` auf Knoten durchzuführen, auf denen Betriebssysteme ausgeführt werden, für die die Agentsoftware nicht auf Ihrem OpsWorks for Puppet Enterprise-Server vorinstalliert ist. Detaillierte Informationen dazu, wie Sie `puppet-agent` auf Ihrem Server für Knoten mit anderen Betriebssystemen bereitstellen, finden Sie unter Installing agents in der Puppet Enterprise-Dokumentation.

Weitere Informationen dazu, wie Sie Ihrem Puppet-Master automatisch Knoten zuordnen, indem Sie Benutzerdaten für die EC2-Instance angeben, finden Sie unter Automatisches Hinzufügen von Knoten in OpsWorks Puppet Enterprise.

Führen Sie `associateNode()` API-Aufrufe aus

Nachdem Sie Knoten durch Installation hinzugefügt haben`puppet-agent`, senden die Knoten Zertifikatssignieranfragen (CSRs) an den OpsWorks for Puppet Enterprise-Server. Sie können den CSRs in der Puppet-Konsole anzeigen. Weitere Informationen über Knoten-CSRs finden Sie unter Managing certificate signing requests in der Puppet Enterprise-Dokumentation. Beim Ausführen des `associateNode()` API-Aufrufs OpsWorks für Puppet Enterprise werden Knoten-CSRs verarbeitet und der Knoten Ihrem Server zugeordnet. Im Folgenden finden Sie ein Beispiel dafür, wie Sie diesen API-Aufruf verwenden, AWS CLI um einen einzelnen Knoten zuzuordnen. Sie benötigen die PEM-formatierte CSR, die der Knoten sendet. Diese erhalten Sie von der Puppet-Konsole.

```
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'
```

Weitere Informationen zum automatischen Hinzufügen von Knoten mit `associateNode()` finden Sie unter Automatisches Hinzufügen von Knoten in OpsWorks Puppet Enterprise.

Überlegungen beim Hinzufügen von lokalen Knoten

Nach `puppet-agent` der Installation auf Ihren lokalen Computern oder virtuellen Maschinen können Sie eine von zwei Methoden verwenden, um lokale Knoten Ihrem OpsWorks for Puppet Enterprise-Master zuzuordnen.

- Wenn ein Knoten die Installation des AWS-SDKs, der AWS CLI oder von AWS Tools for PowerShell unterstützt, können Sie die empfohlene Methode für das Zuordnen von Knoten verwenden: Führen Sie den API-Aufruf `associateNode()` aus. Das Starterkit, das Sie herunterladen, wenn Sie zum ersten Mal einen Master OpsWorks für Puppet Enterprise erstellen, zeigt, wie Sie Knoten mithilfe von Tags Rollen zuweisen. Durch Festlegen von vertrauenswürdigen Fakten in der Zertifikatssignierungsanforderung können Sie Tags gleichzeitig mit dem Zuordnen von Knoten zum Puppet-Master anwenden. Das Demo-Kontroll-Repository aus dem Starter Kit ist beispielsweise so konfiguriert, dass das Tag `pp_role` zum Zuweisen von Rollen an Amazon EC2-Instances verwendet wird. Weitere Informationen zum Hinzufügen von Tags als vertrauenswürdige Fakten zu einer Zertifikatssignierungsanforderung finden Sie unter Extension requests (permanent certificate data) in der Puppet-Plattform-Dokumentation.
- Wenn der Knoten keine AWS Verwaltungs- oder Entwicklungstools ausführen kann, können Sie ihn trotzdem bei Ihrem OpsWorks for Puppet Enterprise-Master registrieren, genauso wie Sie ihn bei einem nicht verwalteten Puppet Enterprise-Master registrieren würden. Wie

in diesem Thema erwähnt, wird bei der Installation eine CSR an den Master OpsWorks für Puppet Enterprise `puppet-agent` gesendet. Ein autorisierter Puppet-Benutzer kann die Zertifikatsignierungsanforderung entweder manuell signieren oder die automatische Signierung konfigurieren, indem er die auf dem Puppet-Master gespeicherte Datei `autosign.conf` bearbeitet. Weitere Informationen zum Konfigurieren der automatischen Signierung und zum Bearbeiten der Datei `autosign.conf` finden Sie unter SSL configuration: autosigning certificate requests in der Puppet-Dokumentation.

Um lokale Knoten mit einem Puppet-Master zu verknüpfen und dem Puppet-Master zu erlauben, alle CSRs zu akzeptieren, gehen Sie auf der Puppet Enterprise-Konsole wie folgt vor. Der Parameter zur Steuerung des Verhaltens ist `puppet_enterprise::profile::master::allow_unauthenticated_ca`.

> **Important**
>
> Aus Sicherheitsgründen ist es nicht empfehlenswert, den Puppet-Master in die Lage zu versetzen, selbstsignierte CSRs oder alle CSRs zu akzeptieren. Standardmäßig wird ein Puppet-Master für die ganze Welt zugänglich, wenn nicht authentifizierte CSRs zugelassen werden. Wenn Sie den Upload von Zertifikatsanforderungen standardmäßig aktivieren, kann Ihr Puppet-Master anfällig für Denial-of-Service (DoS)-Angriffe werden.

1. Anmeldung bei der Puppet Enterprise-Konsole.
2. Wählen Sie Configure (Konfigurieren), dann Classification (Klassifizierung) und anschließend PE Master (PE-Master) aus und wechseln Sie zur Registerkarte Configuration (Konfiguration).
3. Suchen Sie auf der Registerkarte Classification (Klassifizierung) die Klasse puppet_enterprise::profile::master.
4. Setzen Sie den Wert des Parameters allow_unauthenticated_ca auf true (Wahr).
5. Speichern Sie Ihre Änderungen. Ihre Änderungen werden während der nächsten Puppet-Ausführung angewendet. Sie können 30 Minuten warten, bis die Änderungen wirksam werden (und lokale Knoten hinzugefügt werden), oder Sie können einen Puppet-Lauf manuell im Abschnitt Run (Ausführen) der PE-Konsole starten.

Weitere Informationen

Besuchen Sie die Learn Puppet-Tutorial-Website, um mehr über die Verwendung von Servern OpsWorks für Puppet Enterprise und die Funktionen der Puppet Enterprise-Konsole zu erfahren.

Anmeldung bei der Puppet Enterprise-Konsole

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Nachdem Sie die Anmeldedaten von der Eigenschaftenseite des Puppet-Masters heruntergeladen haben und wenn der Server online ist, melden Sie sich bei der Puppet Enterprise-Konsole an. In dieser Anleitung haben wir Sie angewiesen, Ihr Steuerungs-Repository anzugeben, das Ihre Module enthält, und mindestens einen Knoten hinzuzufügen, der verwaltet werden soll. Auf diese Weise werden Informationen über die Agenten und Knoten in der Konsole angezeigt.

Wenn Sie versuchen, eine Verbindung zur Webseite der Puppet Enterprise-Konsole herzustellen, werden Zertifikatswarnungen in Ihrem Browser angezeigt, bis Sie ein AWS OpsWorks spezifisches, von einer Zertifizierungsstelle signiertes SSL-Zertifikat auf dem Client-Computer installieren, den Sie zur Verwaltung Ihres Puppet-Servers verwenden. Wenn die Warnungen beim Aufrufen der Dashboard-Webseite nicht erscheinen sollen, installieren Sie das SSL-Zertifikat, bevor Sie sich anmelden.

Um das SSL-Zertifikat zu installieren AWS OpsWorks

- Wählen Sie das für Ihr System geeignete Zertifikat.
  - Laden Sie für Linux- oder macOS-basierte Systeme die Datei mit der PEM-Dateinamenerweiterung vom folgenden Amazon S3 S3-Speicherort herunter: https://s3.amazonaws.com/ opsworks-cm-us-east -1- prod-default-assets /misc/ -2016-root.pem. opsworks-cm-ca

> **Note**
>
> Laden Sie zusätzlich eine neuere PEM-Datei vom folgenden Speicherort herunter:. https://s3.amazonaws.com/opsworks-cm-us-east-1-prod-default-assets/misc/opsworks-cm-ca-2020-root.pem Da OpsWorks Puppet Enterprise derzeit seine Stammzertifikate erneuert, müssen Sie sowohl alten als auch neuen Zertifikaten vertrauen.

Weitere Informationen zur Verwaltung von SSL-Zertifikaten auf macOS findest du unter Informationen zu einem Zertifikat in Keychain Access auf dem Mac auf der Apple Support-Website abrufen.

- Für Windows-Systeme laden Sie die Datei mit der Dateinamenerweiterung P7B vom folgenden Amazon S3 S3-Speicherort herunter: https://s3.amazonaws.com/ opsworks-cm-us-east -1 - / misc/ -2016-root.p7b. prod-default-assets opsworks-cm-ca

  Weitere Informationen zur Installation eines SSL-Zertifikats unter Windows finden Sie unter Vertrauenswürdige Stammzertifikate verwalten auf Microsoft TechNet.

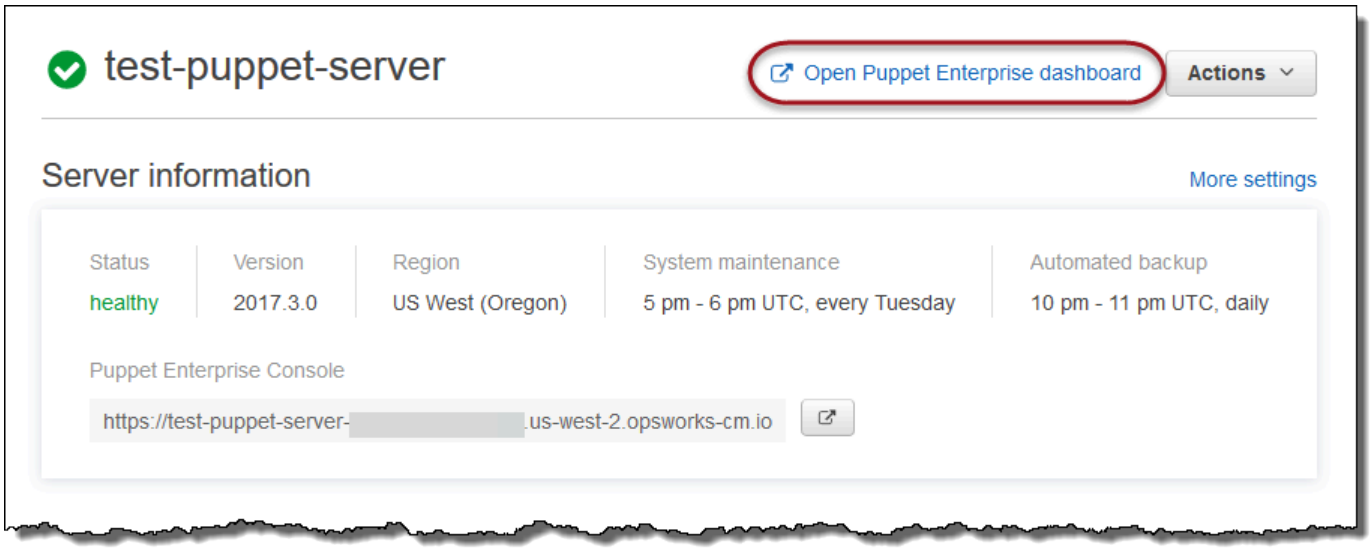
> **Note**
>
> Laden Sie außerdem eine neuere P7B-Datei vom folgenden Speicherort herunter:. https://s3.amazonaws.com/opsworks-cm-us-east-1-prod-default-assets/misc/opsworks-cm-ca-2020-root.p7b Da OpsWorks Puppet Enterprise derzeit seine Stammzertifikate erneuert, müssen Sie sowohl alten als auch neuen Zertifikaten vertrauen.

Nachdem Sie das clientseitige SSL-Zertifikat installiert haben, können Sie sich an der Puppet Enterprise-Konsole anmelden, ohne dass Warnmeldungen angezeigt werden.

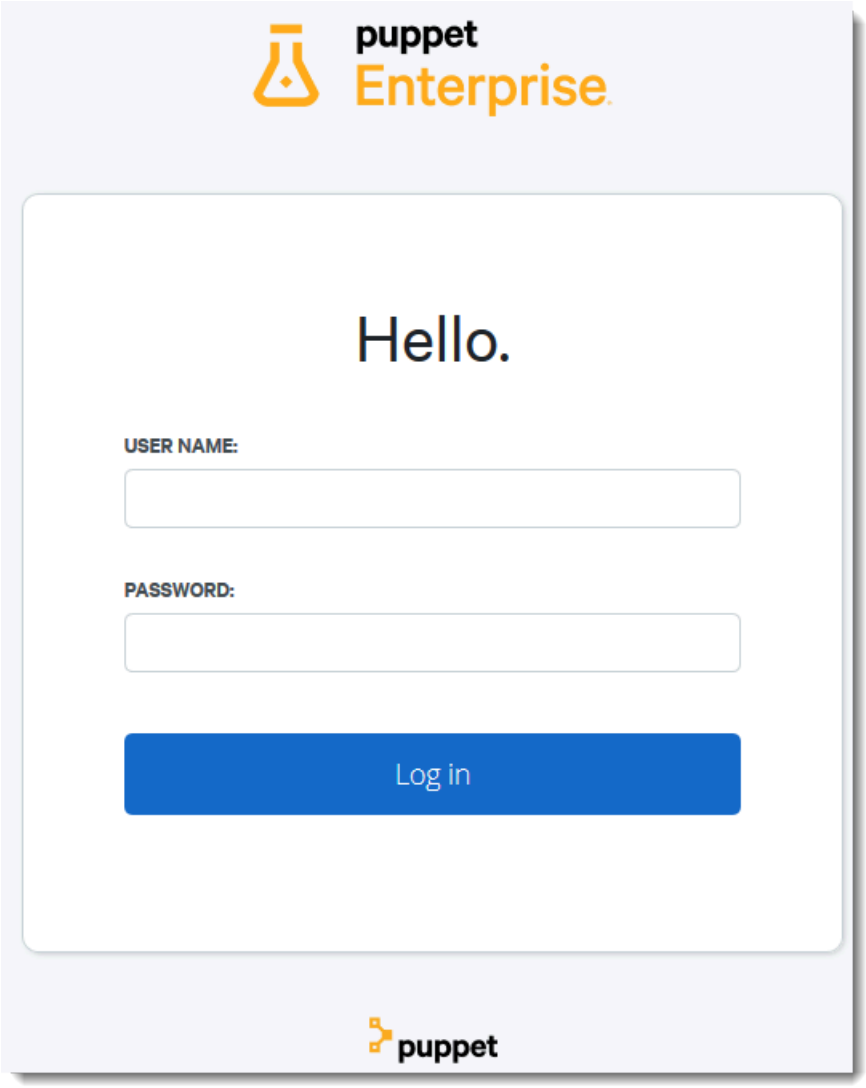
Anmeldung bei der Puppet Enterprise-Konsole

1. Entpacken und öffnen Sie die Puppet Enterprise-Anmeldeinformationen, die Sie unter Voraussetzungen heruntergeladen haben. Sie benötigen diese Anmeldeinformationen, um sich anzumelden.
2. Öffnen Sie im die AWS Management ConsoleEigenschaftenseite für Ihren Puppet-Server.

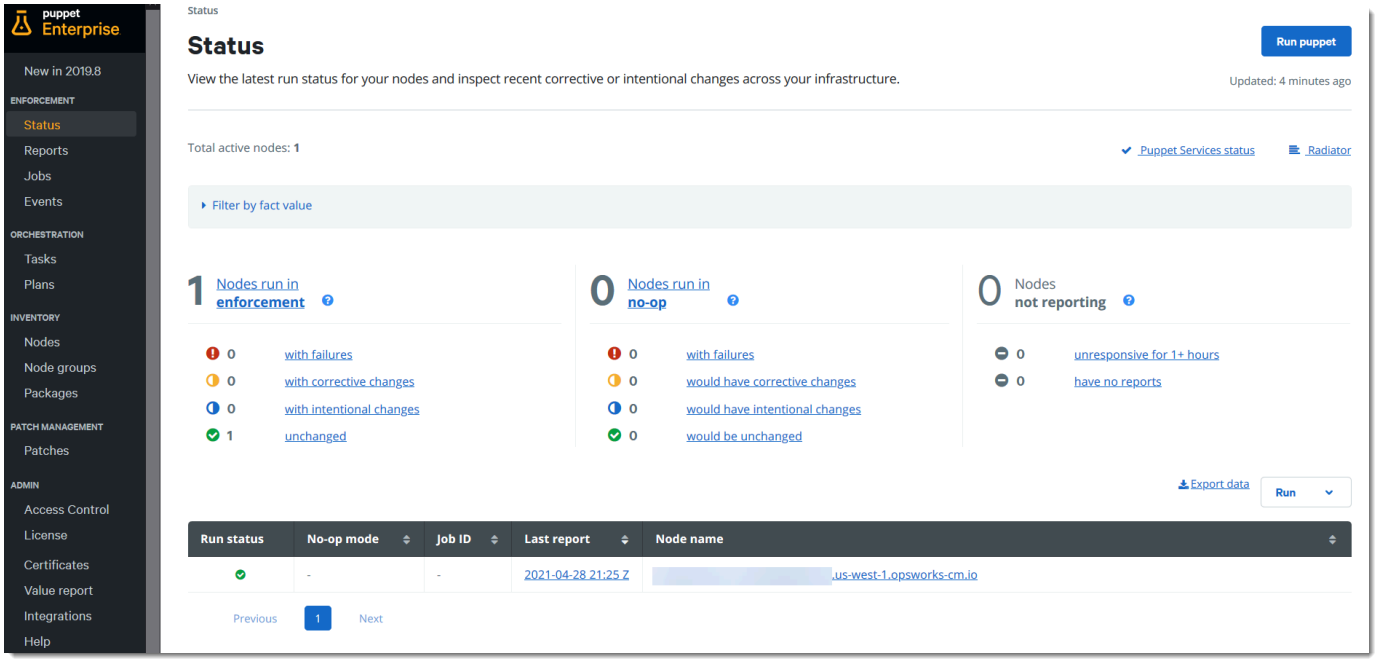
3. Wählen Sie auf der Seite Properties (Eigenschaften) oben rechts Open Puppet Enterprise console (Puppet Enterprise-Konsole öffnen) aus.



4. Melden Sie sich mit den Anmeldeinformationen aus Schritt 1 an.



5. In der Puppet Enterprise-Konsole werden detaillierte Informationen zu den verwalteten Knoten, Modul-Ausführungsfortschritten und Ereignissen, Compliance-Ebenen der Knoten und vieles mehr angezeigt. Weitere Informationen zu den Funktionen der Puppet Enterprise-Konsole und deren Verwendung finden Sie in der Puppet Enterprise-Dokumentation unter Knoten verwalten.



Knoten gruppieren und klassifizieren

Bevor Sie die gewünschte Konfiguration Ihrer Knoten angeben, indem Sie Klassen darauf anwenden, gruppieren Sie die Knoten entsprechend ihrer Rollen in Ihrem Unternehmen oder ihren gemeinsamen Eigenschaften. Für das Gruppieren und Klassifizieren von Knoten sind die folgenden allgemeinen Aufgaben erforderlich. Sie können diese Aufgaben unter Verwendung der PE-Konsole ausführen. Detaillierte Informationen zur Gruppierung und Klassifizierung Ihrer Knoten finden Sie unter Grouping and classifying nodes in der Puppet Enterprise-Dokumentation.

1. Knotengruppen erstellen
2. Fügen Sie Knoten, um Gruppen manuell oder automatisch anhand von Regeln, die Sie erstellen, hinzu.
3. Knotengruppen Klassen zuweisen

Zurücksetzen von Administrator- und Benutzerpasswörtern

Informationen zum Ändern des Passworts, mit dem Sie sich bei der Puppet Enterprise-Konsole anmelden, finden Sie in der Puppet Enterprise-Dokumentation unter Zurücksetzen des Administratorkennworts für die Konsole.

Standardmäßig werden Benutzer nach zehn Anmeldeversuchen aus der Puppet-Konsole gesperrt. Weitere Informationen zum Zurücksetzen der Benutzerkennwörter im Falle einer Sperre finden Sie unter Password endpoints in der Puppet Enterprise-Dokumentation.

Optional: AWS CodeCommit Als Puppet R10k Remote Control Repository verwenden

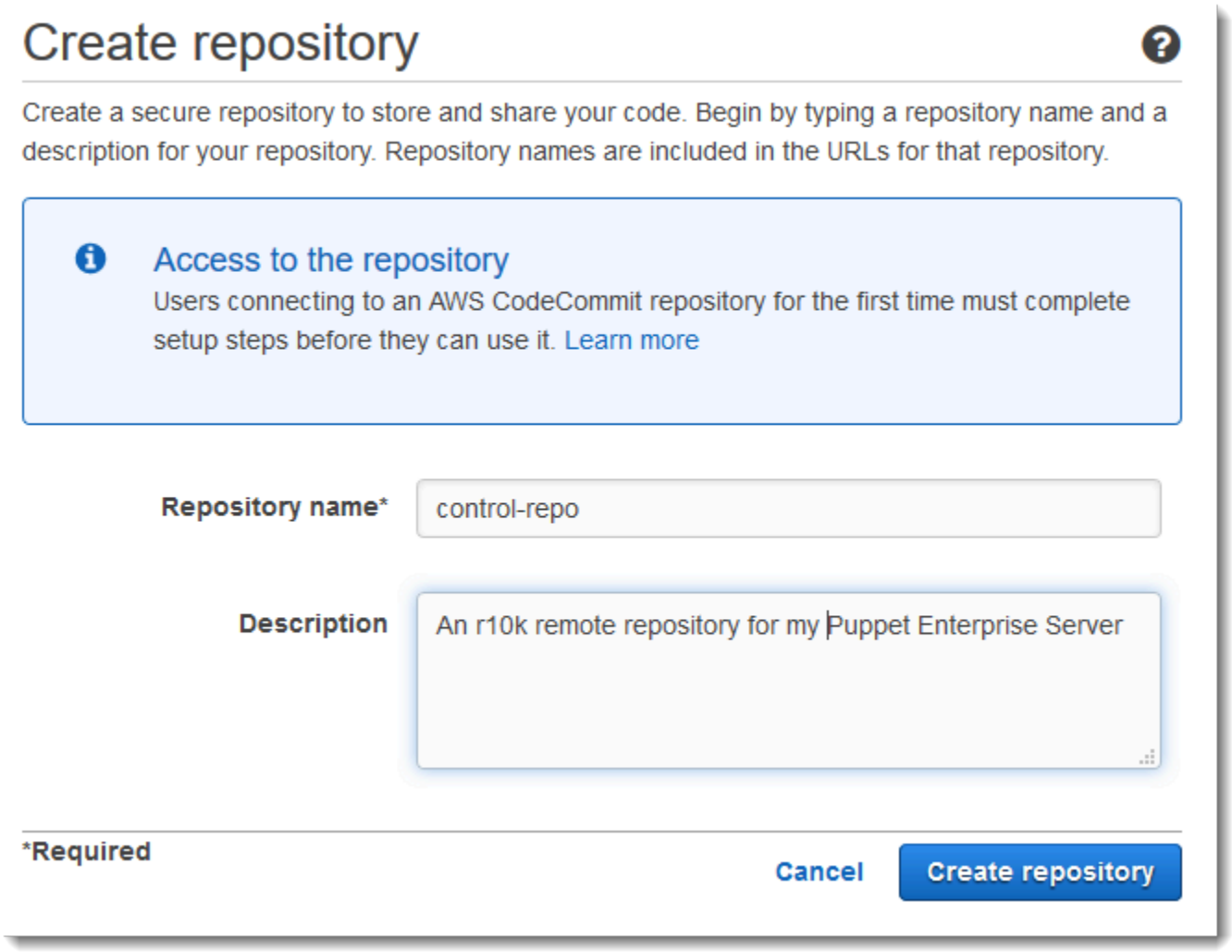
> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können ein neues Repository erstellen AWS CodeCommit, indem Sie es verwenden und es als Ihr R10k-Fernsteuerungs-Repository verwenden. Um die Schritte in diesem Abschnitt ausführen und mit einem CodeCommit Repository arbeiten zu können, benötigen Sie einen Benutzer, der über die in der AWSCodeCommitReadOnlyverwalteten Richtlinie vorgesehenen Berechtigungen verfügt.

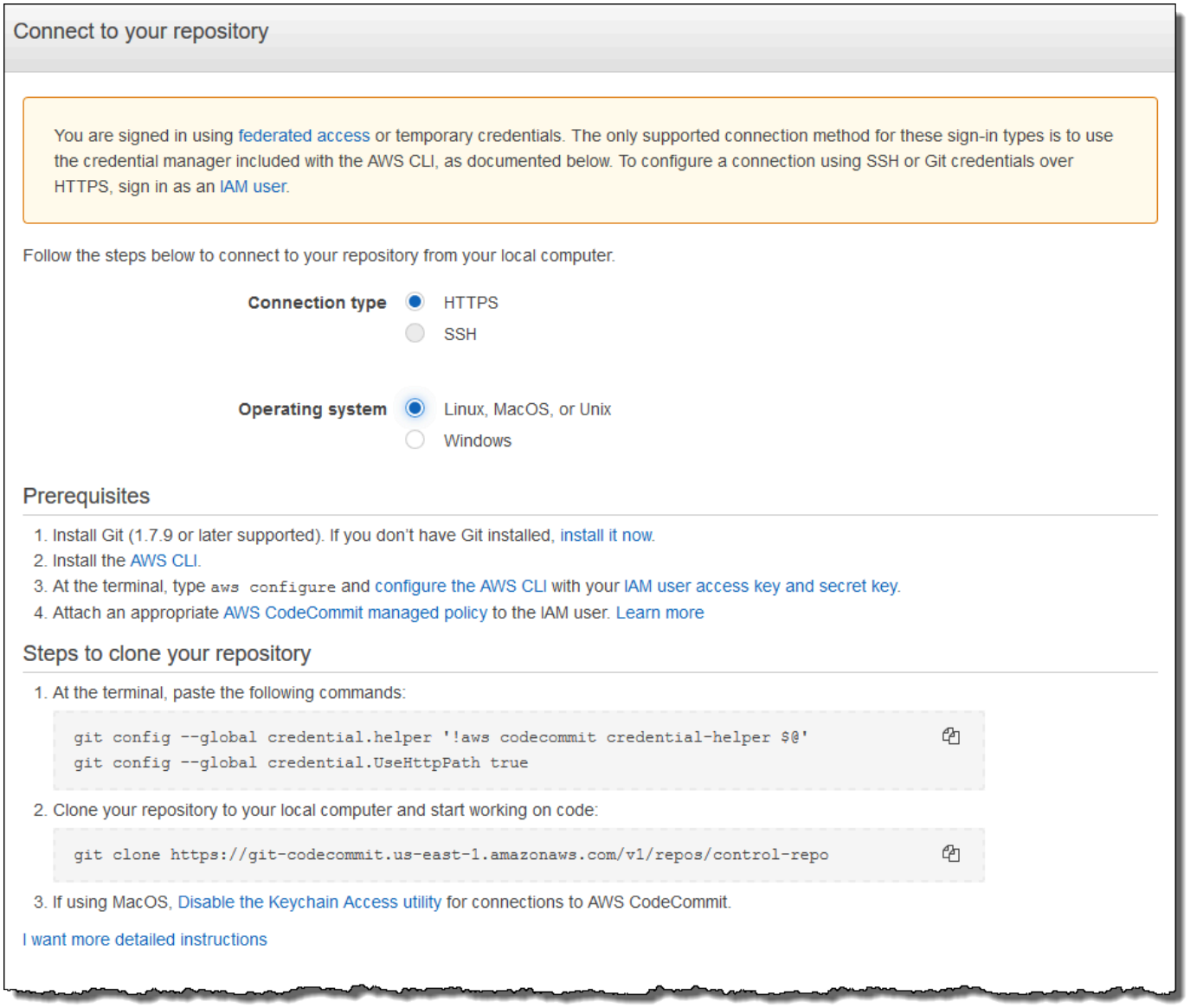
Themen

Schritt 1: CodeCommit Als Repository mit einem HTTPS-Verbindungstyp verwenden

1. Erstellen Sie in der CodeCommit Konsole ein neues Repository.



2. Wählen Sie Überspringen, um die Einrichtung eines Amazon SNS SNS-Themas zu überspringen.
3. Wählen Sie auf der Seite Code die Option Connect to your repository (Mit Repository verbinden) aus.
4. Wählen Sie auf der Seite Connect to your repository (Mit Repository verbinden) die Option HTTPS als Connection type (Verbindungstyp) aus. Wählen Sie anschließend Ihr Betriebssystem aus.



Im Bereich Steps to clone your repository (Repository schrittweise klonen) sollte Ihre `git clone`-URL etwa wie folgt aussehen: `https://git-codecommit.`*`region`*`.amazonaws.com/v1/repos/control-repo`. Kopieren Sie diese URL an einen leicht zugänglichen Speicherort für die Einrichtung des Puppet-Servers.

5. Schließen Sie die Seite Mit Ihrem Repository verbinden und kehren Sie zum Server-Setup OpsWorks für Puppet Enterprise zurück.
6. Fügen Sie die URL, die Sie in Schritt 4 im Textfeld r10k remote kopiert haben, auf der Seite Configure credentials (Anmeldeinformationen konfigurieren) des Einrichtungsassistenten für den Puppet-Master ein. Tragen Sie nichts in das Feld r10k private key (r10k-Privatschlüssel) ein. Schließen Sie das Erstellen und Starten Ihres Puppet-Masters ab.

7. Fügen Sie in der IAM-Konsole die AWSCodeCommitReadOnlyRichtlinie der Instanzprofilrolle Ihres Puppet-Masters hinzu. Weitere Informationen zum Hinzufügen einer Berechtigungsrichtlinie zu einer IAM-Rolle finden Sie unter Hinzufügen von IAM-Identitätsberechtigungen (Konsole) im IAM-Benutzerhandbuch.
8. Folgen Sie den Schritten unter Einrichtung für HTTPS-Benutzer mithilfe von Git-Anmeldeinformationen im AWS CodeCommit Benutzerhandbuch, um Ihre vorhandenen `control-repo` Inhalte in das neue CodeCommit Repository zu übertragen.
9. Jetzt können Sie weiterhin durcharbeiten. Befolgen Sie hierzu die Anweisungen in the section called “Beenden der Konfiguration” und verwenden Sie das Starter Kit, um den Code Ihrem Puppet-Master bereitzustellen. Nachfolgend finden Sie einen Beispielbefehl.

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

Schritt 2: (Optional) CodeCommit Als Repository mit einem SSH-Verbindungstyp verwenden

Sie können ein AWS CodeCommit R10k-Remotesteuerungs-Repository für die Verwendung der SSH-Schlüsselpaar-Authentifizierung konfigurieren. Die folgenden Voraussetzungen müssen vervollständigt werden, bevor Sie mit diesem Vorgang beginnen.

- Sie müssen Ihren OpsWorks for Puppet Enterprise-Server mit einem HTTPS-Steuerungs-Repository gestartet haben, wie im vorherigen Abschnitt beschrieben,. the section called “Schritt 1: CodeCommit Als Repository mit einem HTTPS-Verbindungstyp verwenden” Dies muss zuerst abgeschlossen werden, damit Sie die erforderliche Konfiguration auf den Puppet-Master hochladen können.
- Stellen Sie sicher, dass Sie einen Benutzer haben, dem die AWSCodeCommitReadOnlyverwaltete Richtlinie angehängt ist. Weitere Informationen zum Erstellen eines Benutzers finden Sie unter Erstellen eines IAM-Benutzers in Ihrem AWS Konto im IAM-Benutzerhandbuch.
- Erstellen und verknüpfen Sie einen SSH-Schlüssel mit Ihrem -Benutzer. Folgen Sie den Anweisungen zum Erstellen eines öffentlichen/privaten key pair mit `ssh-keygen` in Schritt 3: Anmeldeinformationen unter Linux, macOS oder Unix konfigurieren im AWS CodeCommit Benutzerhandbuch.

1. Führen Sie in einer AWS CLI Sitzung den folgenden Befehl aus, um den Inhalt der privaten Schlüsseldatei in den AWS Systems Manager Parameter Store hochzuladen. Ihr OpsWorks

for Puppet Enterprise-Server fragt diesen Parameter ab, um eine erforderliche Zertifikatsdatei abzurufen. Ersetzen Sie die *-Datei mit dem privaten Schlüssel* durch den Pfad zu Ihrer Datei mit dem privaten SSH-Schlüssel.

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

2. Fügen Sie Ihrem Puppet-Master Systems Manager Parameter Store-Berechtigungen hinzu.

   a. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.

   b. Wählen Sie im linken Navigationsbereich Roles aus.

   c. Wählen Sie „Zwei Rollen“ aws-opsworks-cm-ec.

   d. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Option Attach policies (Richtlinien anfügen) aus.

   e. Geben Sie im Suchfeld **`AmazonSSMManagedInstanceCore`** ein.

   f. Wählen Sie in den Suchergebnissen ManagedInstanceCoreAmazonSSM aus.

   g. Wählen Sie Richtlinie anfügen aus.

3. Erstellen Sie das Manifest für die Konfigurationsdatei. Wenn Sie das `control-repo-example` Repository, welches Sie im Starter Kit finden, verwenden, erstellen Sie die folgenden Dateien an den Orten aus dem Beispiel-Repository. Andernfalls erstellen Sie sie gemäß Ihrer eigenen Steuerungs-Repository-Struktur. Ersetzen Sie den *IAM_USER_SSH_KEY*-Wert mit der SSH-Schlüssel-ID, die Sie als Voraussetzung für diesen Ablauf erstellt haben.

```
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. Pushen Sie Ihr Kontroll-Repository auf. CodeCommit Führen Sie die folgenden Befehle aus, um die neue Manifestdatei in Ihr Repository zu schieben.

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

5. Stellen Sie die Manifestdateien bereit. Führen Sie die folgenden Befehle aus, um die aktualisierte Konfiguration auf Ihrem OpsWorks for Puppet Enterprise-Server bereitzustellen. Ersetzen Sie *STARTER_KIT_DIRECTORY* mit dem Pfad zu Ihren Puppet-Konfigurationsdateien.

```
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. Aktualisieren Sie die Klassifizierung des OpsWorks for Puppet Enterprise-Servers. Standardmäßig wird der Puppet-Agent alle 30 Minuten auf Knoten (einschließlich dem Master) ausgeführt. Um Wartezeiten zu vermeiden, können Sie den Agenten manuell auf dem Puppet-Master ausführen. Das Ausführen des Agent nimmt die neue Manifestdatei auf.

   a. Anmeldung bei der Puppet Enterprise-Konsole.

   b. Wählen Sie Klassifizierung.

   c. Erweitern Sie PE Infrastructure.

   d. Wählen Sie PE-Master.

   e. Geben Sie auf der Registerkarte Konfiguration **`profile::codecommit`** unter Neue Klasse hinzufügen ein.

      Die neue Klasse `profile::codecommit` wird möglicherweise nicht sofort nach der Ausführung von `puppet-code deploy` angezeigt. Wählen Sie Aktualisieren auf dieser Seite, wenn es nicht angezeigt wird.

   f. Wählen Sie Klasse hinzufügen und wählen Sie dann Commit 1 ändern.

   g. Führen Sie den Puppet-Agent manuell auf dem OpsWorks for Puppet Enterprise-Server aus. Wählen Sie Knoten, wählen Sie Ihren Server aus der Liste, wählen Sie Run Puppet, und wählen Sie dann Run.

7. Ändern Sie in der Puppet-Enterprise-Konsole die Repository-URL auf die Verwendung von SSH anstelle von HTTPS. Die Konfiguration, die Sie in diesen Schritten vornehmen, wird während des Sicherungs- und Wiederherstellungsvorgangs OpsWorks für Puppet Enterprise gespeichert, sodass Sie die Repository-Konfiguration nach Wartungsarbeiten nicht manuell ändern müssen.

   a. Wählen Sie Klassifizierung.

   b. Erweitern Sie PE Infrastructure.

   c. Wählen Sie PE-Master.

   d. Klicken Sie in der Konfigurierungs-Registerkarte auf `puppet_enterprise::profile::master`-Klasse.

   e. Wählen Sie Bearbeiten neben dem `r10k_remote`-Parameter.

   f. Ersetzen Sie die HTTPS-URL durch die SSH-URL für Ihr Repository und wählen Sie dann Commit 1 ändern.

   g. Führen Sie den Puppet-Agent manuell auf dem OpsWorks for Puppet Enterprise-Server aus. Wählen Sie Knoten, wählen Sie Ihren Server aus der Liste, wählen Sie Run Puppet, und wählen Sie dann Run.

Erstellen Sie einen AWS OpsWorks for Puppet Enterprise Master mit AWS CloudFormation

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks for Puppet Enterprise ermöglicht es Ihnen, einen Puppet Enterprise-Server in auszuführen. AWS Sie können in etwa 15 Minuten einen Puppet Enterprise Master-Server einrichten.

Ab dem 3. Mai 2021 speichert Puppet Enterprise einige Puppet Enterprise-Serverattribute in. OpsWorks AWS Secrets Manager Weitere Informationen finden Sie unter Integration in AWS Secrets Manager.

Die folgende exemplarische Vorgehensweise hilft Ihnen beim Erstellen eines Puppet-Masters OpsWorks für Puppet Enterprise, indem Sie einen Stack-In erstellen. AWS CloudFormation

Themen

Voraussetzungen

Bevor Sie einen neuen Puppet-Master erstellen, erstellen Sie außerhalb von Puppet Enterprise die Ressourcen, die Sie OpsWorks für den Zugriff und die Verwaltung Ihres Puppet-Masters benötigen. Weitere Informationen finden Sie unter Voraussetzungen im Abschnitt "Erste Schritte" dieses Handbuchs.

Wenn Sie einen Server erstellen, der eine benutzerdefinierte Domäne verwendet, benötigen Sie eine benutzerdefinierte Domäne, ein benutzerdefiniertes Zertifikat und eine benutzerdefinierten privaten Schlüssel. Sie müssen Werte für alle drei Parameter in Ihrer Vorlage angeben. AWS CloudFormation Weitere Informationen zu den Anforderungen für die `CustomPrivateKey` Parameter `CustomDomainCustomCertificate`, und finden Sie CreateServerin der AWS OpsWorks CM-API-Referenz.

Im Abschnitt OpsWorks -CM der Vorlagenreferenz für das AWS CloudFormation Benutzerhandbuch finden Sie Informationen zu den unterstützten und erforderlichen Werten in der AWS CloudFormation Vorlage, die Sie zum Erstellen Ihres Servers verwenden.

Erstellen eines Puppet Enterprise-Masters in AWS CloudFormation

In diesem Abschnitt wird beschrieben, wie Sie mithilfe einer AWS CloudFormation Vorlage einen Stack erstellen, der einen Masterserver OpsWorks für Puppet Enterprise erstellt. Sie können dies tun, indem Sie die AWS CloudFormation Konsole oder die AWS CLI verwenden. Es steht eine AWS CloudFormation Beispielvorlage zur Verfügung, mit der Sie einen Server-Stack OpsWorks für Puppet Enterprise erstellen können. Achten Sie darauf, die Beispielvorlage mit Ihrem eigenen Servernamen, den IAM-Rollen, dem Instanzprofil, der Serverbeschreibung, der Anzahl der Backup-Aufbewahrungsfristen, den Wartungsoptionen und optionalen Tags zu aktualisieren. Wenn der Server eine benutzerdefinierte Domäne verwendet, müssen Sie Werte für die Parameter `CustomDomain`, `CustomCertificate` und `CustomPrivateKey` in der AWS CloudFormation -Vorlage angeben. Weitere Informationen zu diesen Optionen finden Sie unter the section called “Erstellen Sie einen

Puppet Enterprise Master mithilfe der AWS Management Console" im Abschnitt "Erste Schritte" in diesem Handbuch.

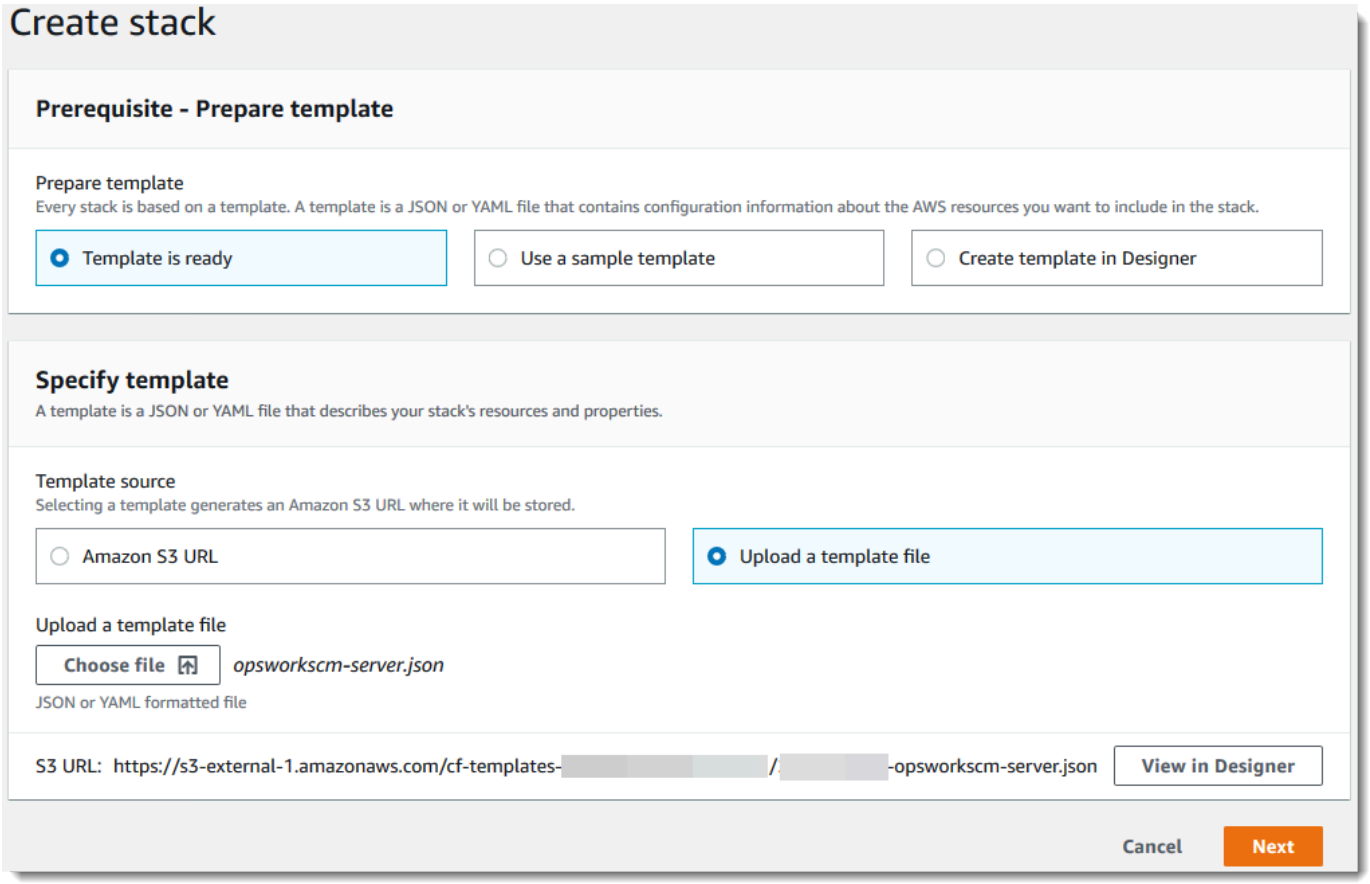
Themen

- Erstellen Sie einen Puppet Enterprise Master mithilfe von AWS CloudFormation (Konsole)
- Erstellen Sie einen Puppet Enterprise Master mithilfe von AWS CloudFormation (CLI)

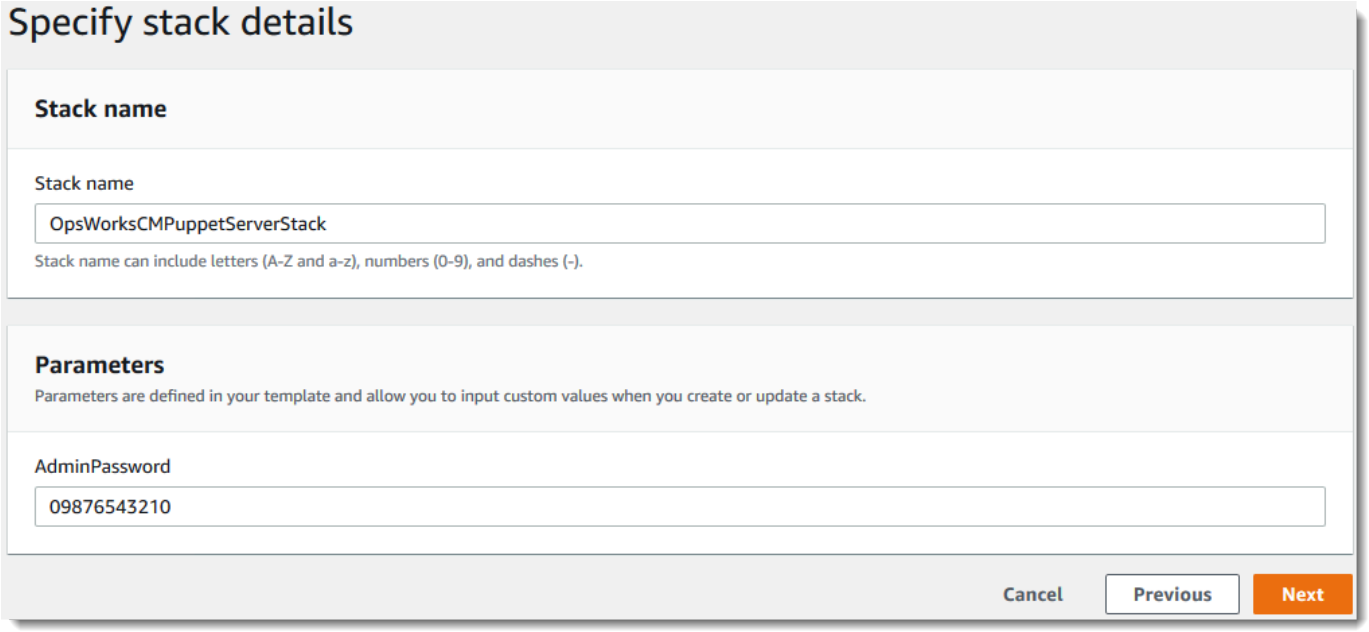
Erstellen Sie einen Puppet Enterprise Master mithilfe von AWS CloudFormation (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie auf der AWS CloudFormation Startseite die Option Stack erstellen aus.
3. Wenn Sie im Schritt Prerequisite - Prepare template (Voraussetzung: Vorbereiten der Vorlage) die AWS CloudFormation -Beispielvorlage verwenden, wählen Sie Template is ready (Vorlage ist bereit) aus.
4. Wählen Sie unter Specify template (Vorlage angeben) die Quelle Ihrer Vorlage aus. Wählen Sie für diese exemplarische Vorgehensweise die Option Eine Vorlagendatei hochladen und laden Sie eine AWS CloudFormation Vorlage hoch, mit der ein Puppet Enterprise-Server erstellt wird. Suchen Sie nach Ihrer Vorlagendatei und klicken Sie dann auf Next (Weiter).

   Eine AWS CloudFormation Vorlage kann entweder im YAML- oder im JSON-Format vorliegen. Es steht Ihnen eine AWS CloudFormation Beispielvorlage zur Verfügung. Achten Sie darauf, die Beispielwerte durch Ihre eigenen zu ersetzen. Sie können den AWS CloudFormation Vorlagendesigner verwenden, um eine neue Vorlage zu erstellen oder eine bestehende zu validieren. Weitere Informationen zu diesem Verfahren finden Sie unter Übersicht über die AWS CloudFormation Designer-Oberfläche im AWS CloudFormation -Benutzerhandbuch.



5. Geben Sie auf der Seite Specify stack details (Stack-Details angeben) einen Namen für den Stack ein. Dies ist nicht dasselbe wie der Name Ihres Servers; es ist nur ein Stack-Name. Geben Sie im Bereich Parameters (Parameter) ein Administratorpasswort ein, das für die Anmeldung auf der Puppet Enterprise-Konsolenwebseite verwendet werden soll. Das Passwort muss zwischen 8 und 32 ASCII-Zeichen lang sein. Wählen Sie Weiter aus.



6. Auf der Seite „Optionen“ können Sie dem Server, den Sie mit dem Stack erstellen, Tags hinzufügen und eine IAM-Rolle für die Erstellung von Ressourcen auswählen, falls Sie noch keine IAM-Rolle zur Verwendung in Ihrer Vorlage angegeben haben. Wenn Sie alle Optionen angegeben haben, wählen Sie Next (Weiter) aus. Weitere Informationen zu erweiterten Optionen wie Rollback-Triggern finden Sie im Benutzerhandbuch unter Setting AWS CloudFormation Stack Options.AWS CloudFormation
7. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Wenn Sie bereit sind, den Server-Stack zu erstellen, wählen Sie Create (Erstellen) aus.

   Während Sie darauf warten, den Stack AWS CloudFormation zu erstellen, sehen Sie sich den Status der Stack-Erstellung an. Wenn die Stack-Erstellung fehlschlägt, überprüfen Sie die Fehlermeldungen in der Konsole, die Sie bei der Fehlerbehebung unterstützen. Weitere Informationen zur Fehlerbehebung bei AWS CloudFormation -Stacks finden Sie im Abschnitt zur Fehlerbehebung im AWS CloudFormation -Benutzerhandbuch.

   Wenn die Servererstellung abgeschlossen ist, ist Ihr Puppet-Master auf der OpsWorks Startseite von Puppet Enterprise mit dem Status Online verfügbar. Nachdem sich der Server online befindet, ist die Puppet Enterprise-Konsole auf der Server-Domäne mit einer URL mit folgendem Format verfügbar: `https://your_server_name-randomID.region.opsworks-cm.io`.

> **Note**
>
> Wenn Sie eine benutzerdefinierte Domäne, ein Zertifikat und einen privaten Schlüssel für Ihren Server angegeben haben, erstellen Sie im DNS-Management-Tool Ihres Unternehmens einen CNAME-Eintrag, der Ihre benutzerdefinierte Domäne dem Endpunkt zuordnet, der OpsWorks für Puppet Enterprise automatisch für den Server generiert wurde. Sie können weder den Server verwalten noch eine Verbindung mit der Puppet Enterprise-Managementwebsite für den Server herstellen, bis Sie den generierten Endpunkt Ihrem benutzerdefinierten Domänenwert zuordnen.
> Um den generierten Endpunktwert abzurufen, führen Sie den folgenden AWS CLI Befehl aus, nachdem Ihr Server online ist:
>
> ```
> aws opsworks describe-servers --server-name server_name
> ```

Erstellen Sie einen Puppet Enterprise Master mithilfe von AWS CloudFormation (CLI)

Wenn auf Ihrem lokalen Computer das noch nicht ausgeführt wird AWS CLI, laden Sie es herunter und installieren Sie es, AWS CLI indem Sie den Installationsanweisungen im AWS-Benutzerhandbuch für die Befehlszeilenschnittstelle folgen. In diesem Abschnitt werden nicht alle Parameter beschrieben, die Sie mit dem Befehl create-stack verwenden können. Weitere Informationen zu den create-stack-Parametern finden Sie unter create-stack in der AWS CLI - Referenz.

1. Stellen Sie sicher, dass Sie die Schritte Voraussetzungen zur Erstellung eines Masters OpsWorks für Puppet Enterprise abgeschlossen haben.
2. Erstellen Sie eine Servicerolle und ein Instanzprofil. AWS OpsWorks stellt eine AWS CloudFormation Vorlage bereit, mit der Sie beide erstellen können. Führen Sie den folgenden AWS CLI Befehl aus, um einen AWS CloudFormation Stack zu erstellen, der die Servicerolle und das Instanzprofil für Sie erstellt.

   ```
   aws cloudformation create-stack --stack-name OpsWorksCMRoles --template-url
    https://s3.amazonaws.com/opsworks-cm-us-east-1-prod-default-assets/misc/opsworks-
   cm-roles.yaml --capabilities CAPABILITY_NAMED_IAM
   ```

Suchen Sie nach AWS CloudFormation Abschluss der Erstellung des Stacks die ARNs der Servicerollen in Ihrem Konto und kopieren Sie sie.

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

Suchen Sie in den Ergebnissen des Befehls `list-roles` nach den Einträgen der Servicerolle und des Instance-Profils. Diese sehen etwa wie folgt aus. Notieren Sie sich die ARNs der Servicerolle und des Instanzprofils und fügen Sie sie der AWS CloudFormation Vorlage hinzu, mit der Sie Ihren Puppet-Master-Server-Stack erstellen.

```
{
    "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. Erstellen Sie den Master OpsWorks für Puppet Enterprise, indem Sie den create-stack Befehl erneut ausführen.

   - Ersetzen Sie *stack_name* durch den Namen Ihres Stacks. Dies ist der Name des AWS CloudFormation Stacks, nicht Ihres Puppet-Masters. Der Name des Puppet-Masters ist der Wert von `ServerName` in Ihrer AWS CloudFormation -Vorlage.
   - Ersetzen Sie *template* durch den Pfad zu Ihrer Vorlagendatei und die Erweiterung *yaml or json* mit `.yaml` bzw. `.json` (wie zutreffend).
   - Die Werte für `--parameters` entsprechen EngineAttributesder CreateServerAPI. Für Puppet sind die folgenden Attribute vom Benutzer angegebene Engine-Attribute für die Erstellung eines Servers. r10k-Engine-Attribute verbinden Ihren Puppet-Master mit einem Code-Repository, um die Umgebungskonfiguration des Servers zu verwalten. Weitere Informationen zu r10k-Engine-Attributen finden Sie unter Managing code with r10k (Verwalten von Code mit r10k) in der Puppet Enterprise-Dokumentation.
     - `PUPPET_ADMIN_PASSWORD`, ein Administratorpasswort für die Anmeldung auf der Puppet Enterprise-Konsolenwebseite. Das Passwort muss 8 bis 32 ASCII-Zeichen umfassen und mindestens einen Großbuchstaben, einen Kleinbuchstaben, eine Zahl und ein Sonderzeichen enthalten.
     - `PUPPET_R10K_REMOTE`, die URL Ihres Steuerungs-Repositorys (z. B. ssh://git@your.git-repo.com:user/control-repo.git). Durch das Angeben eines r10k-Remote-Repositorys wird der TCP-Port 8170 geöffnet.
     - `PUPPET_R10K_PRIVATE_KEY`. Wenn Sie ein privates Git-Repository verwenden, fügen Sie PUPPET_R10K_PRIVATE_KEY hinzu, um eine SSH-URL und einen PEM-codierten privaten SSH-Schlüssel anzugeben.

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

Im Folgenden wird ein Beispiel gezeigt.

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

Im folgenden Beispiel werden R10k-Engine-Attribute als Parameter angegeben, wenn sie nicht in der Vorlage bereitgestellt werden. AWS CloudFormation Eine Beispielvorlage, die die r10k-Engine-Attribute enthält, `puppet-server-param-attributes.yaml`, ist bei den AWS CloudFormation -Beispielvorlagen enthalten.

```
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)"
```

Im folgenden Beispiel werden r10k-Engine-Attribute und ihre Werte in der AWS CloudFormation -Vorlage festgelegt. Der Befehl muss nur auf die Vorlagendatei verweisen. Die Vorlage, die als Wert für `--template-body` angegeben ist, `puppet-server-in-file-attributes.yaml`, ist bei den AWS CloudFormation -Beispielvorlagen enthalten.

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

4. (Optional) Um den Status der Stack-Erstellung anzuzeigen, führen Sie den folgenden Befehl aus.

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

5. Wenn die Stack-Erstellung abgeschlossen ist, fahren Sie mit dem nächsten Abschnitt fort, the section called “Beenden der Konfiguration”. Wenn die Stack-Erstellung fehlschlägt, überprüfen Sie die Fehlermeldungen in der Konsole, die Sie bei der Fehlerbehebung unterstützen. Weitere Informationen zur Behebung von Fehlern in AWS CloudFormation Stacks finden Sie unter Problembehandlung im AWS CloudFormation Benutzerhandbuch.

Aktualisieren Sie einen OpsWorks for Puppet Enterprise Server, um eine benutzerdefinierte Domain zu verwenden

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Abschnitt wird beschrieben, wie ein vorhandener Server OpsWorks für Puppet Enterprise aktualisiert wird, sodass er eine benutzerdefinierte Domäne und ein benutzerdefiniertes Zertifikat verwendet, indem ein Backup des Servers verwendet wird, um einen neuen Server zu erstellen. Im Wesentlichen kopieren Sie einen vorhandenen Server OpsWorks für Puppet Enterprise 2.0, indem Sie einen neuen Server aus einem Backup erstellen und dann den neuen Server so konfigurieren, dass er eine benutzerdefinierte Domäne, ein benutzerdefiniertes Zertifikat und einen privaten Schlüssel verwendet.

Themen

Voraussetzungen

Im Folgenden sind die Voraussetzungen für die Aktualisierung eines vorhandenen Servers OpsWorks für Puppet Enterprise aufgeführt, sodass er eine benutzerdefinierte Domäne und ein benutzerdefiniertes Zertifikat verwenden kann.

- Auf dem Server, den Sie aktualisieren (oder kopieren) möchten, muss Puppet Enterprise 2019.8.5 ausgeführt werden.
- Legen Sie fest, welche Sicherung Sie zum Erstellen eines neuen Servers verwenden möchten. Sie müssen mindestens eine Sicherung des Servers zur Verfügung haben, den Sie aktualisieren

möchten. Weitere Informationen zu Backups OpsWorks für Puppet Enterprise finden Sie unter. Einen OpsWorks für Puppet Enterprise Server sichern

- Bereiten Sie die Servicerollen- und Instance-Profil-ARNs vor, mit denen Sie den vorhandenen Server erstellt haben, der die Quelle Ihrer Sicherung ist.
- Stellen Sie sicher, dass Sie die neueste Version von AWS CLI ausführen. Weitere Informationen zur Aktualisierung Ihrer AWS CLI Tools finden Sie unter Installation von AWS CLI im AWS-Benutzerhandbuch für die Befehlszeilenschnittstelle.

Einschränkungen

Wenn Sie einen vorhandenen Server aktualisieren, indem Sie einen neuen Server aus einem Backup erstellen, kann der neue Server nicht exakt mit dem vorhandenen Server OpsWorks für Puppet Enterprise identisch sein.

- Sie können dieses Verfahren nur mit dem AWS CLI oder einem der AWS SDKs abschließen. Sie können mit der AWS Management Console keinen neuen Server aus einer Sicherung erstellen.
- Der neue Server kann nicht denselben Namen wie der vorhandene Server in einem Konto und in einer AWS-Region verwenden. Der Name muss sich von dem vorhandenen Server unterscheiden, den Sie als Quelle der Sicherung verwendet haben.
- Knoten, die an den vorhandenen Server angeschlossen wurden, werden nicht vom neuen Server verwaltet. Sie müssen einen der folgenden Schritte ausführen.
  - Weisen Sie verschiedene Knoten zu, da Knoten von nicht von mehr als einem Puppet-Master verwaltet werden können.
  - Migrieren Sie die Knoten vom vorhandenen Server (der Quelle der Sicherung) auf den neuen Server und den neuen benutzerdefinierten Domänenendpunkt. Weitere Informationen zum Migrieren von Knoten finden Sie in der Puppet Enterprise-Dokumentation.

Aktualisieren eines Servers zur Verwendung einer benutzerdefinierten Domäne

Um einen vorhandenen Puppet-Master zu aktualisieren, erstellen Sie eine Kopie davon, indem Sie den Befehl `create-server` ausführen und Parameter hinzufügen, um eine Sicherung, eine benutzerdefinierte Domäne, ein benutzerdefiniertes Zertifikat und einen benutzerdefinierten privaten Schlüssel anzugeben.

1. Wenn Sie in Ihrem `create-server`-Befehl keine Servicerollen- oder Instance-Profil-ARNs zur Verfügung haben, führen Sie die Schritte 1-5 in Erstellen Sie einen Chef Automate-Server mit dem AWS CLI aus, um eine Servicerolle und ein Instance-Profil zu erstellen, die Sie verwenden können.
2. Wenn Sie dies noch nicht getan haben, suchen Sie die Sicherung des vorhandenen Puppet-Master, auf dem Sie einen neuen Server mit einer benutzerdefinierten Domäne erstellen möchten. Führen Sie den folgenden Befehl aus, um Informationen zu allen OpsWorks Backups von Puppet Enterprise in Ihrem Konto und in einer Region anzuzeigen. Notieren Sie sich die ID der Sicherung, die Sie verwenden möchten.

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

3. Erstellen Sie den Server OpsWorks für Puppet Enterprise, indem Sie den `create-server` Befehl ausführen.

   - Der `--engine` Wert ist `PuppetMonolithic`, `--engine-model` ist und `--engine-version` ist `2019` oder`2017`.
   - Der Servername muss innerhalb Ihres AWS Kontos in jeder Region eindeutig sein. Servernamen müssen mit einem Buchstaben beginnen. Danach können Buchstaben, Zahlen und Bindestriche (-) verwendet werden, insgesamt höchstens 40 Zeichen.
   - Verwenden Sie die ARNs des Instance-Profils und der Servicerolle, die Sie in Schritt 3 und 4 kopiert haben.
   - Gültige Instance-Typen sind `c4.large`, `c4.xlarge` und `c4.2xlarge`. Weitere Informationen zu den Spezifikationen dieser Instance-Typen finden Sie unter Instance-Typen im Amazon EC2 EC2-Benutzerhandbuch.
   - Der Parameter `--engine-attributes` ist optional. Wenn Sie kein Puppet-Administratorpasswort festlegen, wird bei der Servererstellung ein Passwort generiert. Wenn Sie `--engine-attributes` hinzufügen, geben Sie für `PUPPET_ADMIN_PASSWORD` ein Administratorpasswort für die Anmeldung auf der Puppet Enterprise-Konsolenwebseite ein. Das Passwort muss zwischen 8 und 32 ASCII-Zeichen lang sein.
   - Ein SSH-Schlüsselpaar ist optional. Es kann Ihnen dabei helfen, sich mit dem Puppet-Master zu verbinden, wenn Sie das Konsolenadministratorpasswort zurücksetzen müssen. Weitere Informationen zum Erstellen eines SSH-Schlüsselpaars finden Sie unter Amazon EC2 EC2-Schlüsselpaare im Amazon EC2 EC2-Benutzerhandbuch.
   - Um eine benutzerdefinierte Domäne zu verwenden, fügen Sie dem Befehl die folgenden Parameter hinzu. Andernfalls generiert der Erstellungsprozess des Puppet-Master automatisch

einen Endpunkt für Sie. Alle drei Parameter sind erforderlich, um eine benutzerdefinierte Domäne zu konfigurieren. Informationen zu zusätzlichen Anforderungen für die Verwendung dieser Parameter finden Sie CreateServerin der AWS OpsWorks CM-API-Referenz.

- `--custom-domain` – Ein optionaler öffentlicher Endpunkt eines Servers, z. B. `https://aws.my-company.com`.
- `--custom-certificate` – Ein PEM-formatiertes HTTPS-Zertifikat. Der Wert kann ein einzelnes, selbstsigniertes Zertifikat oder eine Zertifikatkette sein.
- `--custom-private-key` – Ein privater Schlüssel im PEM-Format für die Verbindung mit dem Server mithilfe von HTTPS. Der private Schlüssel darf nicht verschlüsselt sein; er kann nicht durch ein Passwort oder eine Passphrase geschützt werden.

- Es ist eine wöchentliche Systemwartung erforderlich. Gültige Werte müssen im folgenden Format angegeben werden: `DDD:HH:MM`. Die angegebene Uhrzeit entspricht der Zeitzone UTC (Coordinated Universal Time). Wenn Sie für `--preferred-maintenance-window` keinen Wert angeben, wird ein zufälliger Standardwert mit einem einstündigen Zeitraum an einem Dienstag, Mittwoch oder Freitag festgelegt.
- Gültige Werte für `--preferred-backup-window` müssen in einem der folgenden Formate angegeben werden: `HH:MM` für tägliche Sicherungen oder `DDD:HH:MM` für wöchentliche Sicherungen. Die angegebene Uhrzeit entspricht der Zeitzone UTC. Standardmäßig wird ein zufälliger täglicher Startzeitpunkt festgelegt. Wenn Sie automatische Sicherungen deaktivieren möchten, verwenden Sie stattdessen den Parameter `--disable-automated-backup`.
- Geben Sie für `--security-group-ids` eine oder mehrere Sicherheitsgruppen-IDs, durch Kommata getrennt, ein.
- Geben Sie für `--subnet-ids` eine Subnetz-ID ein.

```
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
```

Im folgenden Beispiel wird ein Puppet-Master erstellt, der eine benutzerdefinierte Domäne verwendet.

```
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 für Puppet Enterprise dauert die Erstellung eines neuen Servers etwa 15 Minuten. Kopieren Sie in der Ausgabe des Befehls `create-server` den Wert des Attributs `Endpoint`. Im Folgenden wird ein Beispiel gezeigt.

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

   Sie sollten die Ausgabe des Befehls `create-server` nicht verwerfen oder die Shell-Sitzung beenden, da die Ausgabe wichtige Informationen enthalten kann, die nicht wiederhergestellt werden können. Um Passwörter und das Starter Kit aus den Ergebnissen von `create-server` zu extrahieren, fahren Sie mit dem nächsten Schritt fort.

5. Wenn Sie sich dafür entschieden haben OpsWorks , dass Puppet Enterprise ein Passwort für Sie generiert, können Sie es mithilfe eines JSON-Prozessors wie jq in einem verwendbaren Format aus den `create-server` Ergebnissen extrahieren. Nachdem Sie jq installiert haben, können Sie die folgenden Befehle ausführen, um das Puppet-Administratorpasswort und das Starter Kit zu extrahieren. Wenn Sie in Schritt 3 kein eigenes Passwort angegeben haben, speichern Sie das extrahierte Administratorpasswort an einem sicheren Speicherort.

```
#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**
>
> Es ist nicht möglich, in der AWS Management Console ein neues Starter Kit für den Puppet-Master zu erstellen. Wenn Sie mit dem einen Puppet-Master erstellen AWS CLI, führen Sie den vorherigen `jq` Befehl aus, um das Base64-kodierte Starterkit in den `create-server` Ergebnissen als ZIP-Datei zu speichern.

6. Falls Sie das Starterkit nicht aus den `create-server` Befehlsergebnissen extrahiert haben, können Sie optional ein neues Starterkit von der Eigenschaftenseite des Servers in der Konsole OpsWorks für Puppet Enterprise herunterladen.
7. Wenn Sie keine benutzerdefinierte Domäne verwenden, fahren Sie mit dem nächsten Schritt fort. Wenn Sie eine benutzerdefinierte Domäne mit dem Server verwenden, erstellen Sie einen CNAME-Eintrag im DNS-Verwaltungstool Ihres Unternehmens, um Ihre benutzerdefinierte Domäne auf den OpsWorks for Puppet Enterprise-Endpunkt zu verweisen, den Sie in Schritt 4 kopiert haben. Sie können einen Server erst dann mit einer benutzerdefinierten Domäne erreichen und sich erst dann bei ihm anmelden, nachdem Sie diesen Schritt ausgeführt haben.
8. Fahren Sie nach Abschluss der Servererstellung mit Konfigurieren des Puppet-Masters mit dem Starter Kit fort.

Weitere Informationen finden Sie unter:

- Erstellen Sie einen Puppet Enterprise Master mit dem AWS CLI
- Einen OpsWorks for Puppet Enterprise Server sichern und wiederherstellen
- CreateServerin der AWS OpsWorks CM-API-Referenz
- create-server in der AWS CLI Befehlsreferenz

Mit Tags auf AWS OpsWorks for Puppet Enterprise Ressourcen arbeiten

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Tags sind Wörter oder Ausdrücke, die in Form von Metadaten zum Identifizieren und Organisieren Ihrer AWS-Ressourcen verwendet werden. OpsWorks Bei Puppet Enterprise kann eine Ressource bis zu 50 vom Benutzer angewendete Tags haben. Jedes Tag besteht aus einem Schlüssel und einem einzelnen optionalen Wert. In Puppet Enterprise können Sie Tags auf die folgenden Ressourcen anwenden: OpsWorks

- OpsWorks für Puppet Enterprise-Server
- Backups von OpsWorks für Puppet Enterprise-Server

Mithilfe von Tags auf AWS Ressourcen können Sie Kosten verfolgen, den Zugriff auf Ressourcen kontrollieren, Ressourcen zur Automatisierung von Aufgaben gruppieren oder Ressourcen nach Zweck oder Lebenszyklusphase organisieren. Weitere Informationen zu den Vorteilen von Tags finden Sie unter AWS-Tagging-Strategien in AWS Answers und unter Verwendung von Kostenzuweisungs-Tags im AWS Billing and Cost Management -Benutzerhandbuch.

Um Tags OpsWorks zur Steuerung des Zugriffs auf Puppet Enterprise-Server oder Backups zu verwenden, erstellen oder bearbeiten Sie Richtlinienerklärungen in AWS Identity and Access Management (IAM). Weitere Informationen finden Sie unter Steuern des Zugriffs auf AWS -Ressourcen mithilfe von Ressourcen-Tags im AWS Identity and Access Management -Benutzerhandbuch.

Wenn Sie Tags auf einen OpsWorks for Puppet Enterprise-Master anwenden, werden die Tags auch auf die Backups des Masters, den Amazon S3 S3-Bucket, in dem die Backups gespeichert sind, die Amazon EC2 EC2-Instance des Masters, die darin gespeichert sind AWS Secrets Manager, und die vom Master verwendete Elastic IP-Adresse angewendet. Tags werden nicht an den AWS

CloudFormation Stack weitergegeben, der zur Erstellung Ihres AWS OpsWorks Puppet-Masters verwendet wird.

Themen

So funktionieren Tags in AWS OpsWorks for Puppet Enterprise

In dieser Version können Sie Tags hinzufügen und verwalten, indem Sie die AWS OpsWorks - CM-API oder die AWS Management Console verwenden. AWS OpsWorks CM versucht auch, Tags, die Sie einem Server hinzufügen, zu den AWS Ressourcen hinzuzufügen, die dem Server zugeordnet sind, einschließlich der EC2-Instance, Secrets in Secrets Manager, Elastic IP-Adresse, Sicherheitsgruppe, S3-Bucket und Backups.

Die folgende Tabelle bietet einen Überblick darüber, wie Sie Tags in OpsWorks Puppet Enterprise hinzufügen und verwalten.

Aktion	Was zu verwenden ist
Fügen Sie Tags zu einem neuen Server OpsWorks für Puppet Enterprise oder einem Backup hinzu, das Sie manuell erstellen.	• Wählen Sie Create Puppet Enterprise server (Puppet Enterprise-Server erstellen) aus und fügen Sie Tags auf der Seite Configure advanced settings (Erweiterte Einstellungen konfigurieren) hinzu. • Wählen Sie auf der Seite Backups für einen vorhandenen Server die Option Backup erstellen und fügen Sie auf der Seite Backup Ihres Puppet Enterprise-Servers erstellen Tags hinzu. • Fügen Sie den Befehlen „`CreateServer`“ oder „`CreateBackup`“ einen `Tags`-Parameter hinzu.

Aktion	Was zu verwenden ist
Anzeigen von Tags auf einer Ressource.	• Wählen Sie auf der Detailseite für Ihren Server im Navigationsbereich die Option Tags aus. • Wählen Sie auf der Seite Backups (Sicherungen) für Ihren Server eine Sicherung und dann Edit backup (Sicherung bearbeiten) aus. • Führen Sie den Befehl `ListTagsForResource` aus.
Fügen Sie einem vorhandenen OpsWorks Puppet Enterprise-Server oder einem Backup Tags hinzu, unabhängig davon, ob das Backup manuell oder automatisch erstellt wurde.	• Wählen Sie auf der Detailseite für Ihren Server im Navigationsbereich die Option Tags und dann Edit (Bearbeiten)aus. • Wählen Sie auf der Seite Backups (Sicherungen) für Ihren Server eine Sicherung und dann Edit backup (Sicherung bearbeiten) aus. • Führen Sie den Befehl `TagResource` aus.
Löschen Sie Tags von einer Ressource.	• Wählen Sie auf der Detailseite für Ihren Server im Navigationsbereich die Option Tags und dann Edit (Bearbeiten)aus. Wählen Sie das X neben den Tags aus, die Sie löschen möchten. • Wählen Sie auf der Seite Backups (Sicherungen) für Ihren Server eine Sicherung und dann Edit backup (Sicherung bearbeiten) aus. Wählen Sie das X neben den Tags aus, die Sie löschen möchten. • Führen Sie den Befehl `UntagResource` aus.

`DescribeServers`- und `DescribeBackups`-Antworten enthalten keine Tag-Informationen. Verwenden Sie die `ListTagsForResource`-API, um Tags anzuzeigen.

Hinzufügen und Verwalten von Tags OpsWorks für Puppet Enterprise (Konsole)

Die Prozeduren in diesem Abschnitt werden in der AWS Management Console durchgeführt.

Wenn Sie Tags hinzufügen, darf ein Tag-Schlüssel nicht leer sein. Der Schlüssel darf maximal 127 Zeichen lang sein und nur Unicode-Buchstaben, Ziffern oder Trennzeichen oder die folgenden Sonderzeichen enthalten: `+ - = . _ : / @`-Tag-Werte sind optional. Sie können einen Tag hinzufügen, der einen Schlüssel, aber keine Werte enthält. Der Wert darf maximal 255 Zeichen lang sein und können nur Unicode-Buchstaben, Ziffern oder Trennzeichen oder die folgenden Sonderzeichen enthalten: `+ - = . _ : / @`.

Themen

Hinzufügen von Tags zu einem neuen Server OpsWorks für Puppet Enterprise (Konsole)

1. Stellen Sie sicher, dass alle Voraussetzungen für die Erstellung eines Masters OpsWorks für Puppet Enterprise erfüllt sind.
2. Führen Sie die Schritte 1-8 in Erstellen Sie einen Puppet Enterprise Master mithilfe der AWS Management Console aus.
3. Nachdem Sie automatische Sicherungseinstellungen festgelegt haben, fügen Sie Tags im Bereich Tags der Seite Configure advanced settings (Erweiterte Einstellungen konfigurieren) hinzu. Sie können maximal 50 Tags hinzufügen. Wenn Sie alle Tags hinzugefügt haben, wählen Sie Next aus.

4. Gehen Sie zu Schritt 11 von Erstellen Sie einen Puppet Enterprise Master mithilfe der AWS Management Console über, und überprüfen Sie die Einstellungen, die Sie für den neuen Server ausgewählt haben.

Hinzufügen von Tags zu einer neuen Sicherung (Konsole)

1. Wählen Sie auf der Startseite OpsWorks für Puppet Enterprise einen vorhandenen Puppet Master aus.
2. Wählen Sie auf der Detailseite des Servers im Navigationsbereich Backups (Sicherungen) aus.
3. Wählen Sie auf der Seite Backups (Sicherungen) die Option Create Backup (Sicherung erstellen) aus.
4. Fügen Sie Tags hinzu. Wählen Sie Create (Erstellen) aus, sobald Sie mit dem Hinzufügen von Tags fertig sind.

Hinzufügen oder Anzeigen von Tags auf einem vorhandenen Server (Konsole)

1. Wählen Sie auf der Startseite OpsWorks für Puppet Enterprise einen vorhandenen Puppet-Master aus, um dessen Detailseite zu öffnen.
2. Wählen Sie im Navigationsbereich Tags oder unten auf der Detailseite die Option View all tags (Alle Tags anzeigen) aus.
3. Wählen Sie auf der Seite Tags die Option Edit (Bearbeiten) aus.
4. Tags auf dem Server hinzufügen oder bearbeiten. Wählen Sie Save (Speichern) aus, wenn Sie fertig sind.

> **Note**
>
> Beachten Sie, dass das Ändern von Tags auf Ihrem Puppet-Master auch Tags für Ressourcen ändert, die dem Server zugewiesen sind, z. B. die EC2-Instance, die Elastic IP-Adresse, die Sicherheitsgruppe, der S3-Bucket und Sicherungen.

Hinzufügen oder Anzeigen von Tags in einer vorhandenen Sicherung (Konsole)

1. Wählen Sie auf der Startseite OpsWorks für Puppet Enterprise einen vorhandenen Puppet-Master aus, um dessen Detailseite zu öffnen.

2. Wählen Sie im Navigationsbereich Backups (Sicherungen) oder im Bereich Recent backups (Zuletzt verwendete Sicherungen) auf der Detailseite die Option View all backups (Alle Sicherungen anzeigen) aus.
3. Wählen Sie auf der Seite Backups (Sicherungen) eine zu verwaltende Sicherung aus, und wählen Sie dann Edit Backups (Sicherung bearbeiten) aus.
4. Tags in der Sicherung hinzufügen oder bearbeiten. Wählen Sie Update (Aktualisieren) aus, wenn Sie fertig sind.

Löschen von Tags aus einem Server (Konsole)

1. Wählen Sie auf der Startseite OpsWorks für Puppet Enterprise einen vorhandenen Puppet-Master aus, um dessen Detailseite zu öffnen.
2. Wählen Sie im Navigationsbereich Tags oder unten auf der Detailseite die Option View all tags (Alle Tags anzeigen) aus.
3. Wählen Sie auf der Seite Tags die Option Edit (Bearbeiten) aus.
4. Wählen Sie das X neben einem Tag aus, um das Tag zu löschen. Wählen Sie Save (Speichern) aus, wenn Sie fertig sind.

> **Note**
>
> Beachten Sie, dass das Ändern von Tags auf Ihrem Puppet-Master auch Tags für Ressourcen ändert, die dem Server zugewiesen sind, z. B. die EC2-Instance, die Elastic IP-Adresse, die Sicherheitsgruppe, der S3-Bucket und Sicherungen.

Löschen von Tags aus einer Sicherung (Konsole)

1. Wählen Sie auf der Startseite OpsWorks für Puppet Enterprise einen vorhandenen Puppet-Master aus, um dessen Detailseite zu öffnen.
2. Wählen Sie im Navigationsbereich Backups (Sicherungen) oder im Bereich Recent backups (Zuletzt verwendete Sicherungen) auf der Detailseite die Option View all backups (Alle Sicherungen anzeigen) aus.
3. Wählen Sie auf der Seite Backups (Sicherungen) eine zu verwaltende Sicherung aus, und wählen Sie dann Edit Backups (Sicherung bearbeiten) aus.

4. Wählen Sie das X neben einem Tag aus, um das Tag zu löschen. Wählen Sie Update (Aktualisieren) aus, wenn Sie fertig sind.

Hinzufügen und Verwalten von Tags OpsWorks für Puppet Enterprise (CLI)

Die Prozeduren in diesem Abschnitt werden in der AWS CLI durchgeführt. Stellen Sie sicher, dass Sie die neueste Version von ausführen, AWS CLI bevor Sie mit der Arbeit mit Tags beginnen. Weitere Informationen zur Installation oder Aktualisierung von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch. AWS CLI

Wenn Sie Tags hinzufügen, darf ein Tag-Schlüssel nicht leer sein. Der Schlüssel darf maximal 127 Zeichen lang sein und nur Unicode-Buchstaben, Ziffern oder Trennzeichen oder die folgenden Sonderzeichen enthalten: `+ - = . _ : / @`-Tag-Werte sind optional. Sie können einen Tag hinzufügen, der einen Schlüssel, aber keine Werte enthält. Der Wert darf maximal 255 Zeichen lang sein und nur Unicode-Buchstaben, Ziffern oder Trennzeichen oder die folgenden Sonderzeichen enthalten: `+ - = . _ : / @`

Themen

- Hinzufügen von Tags zu einem neuen OpsWorks for Puppet Enterprise Server (CLI)
- Hinzufügen von Tags zu einer neuen Sicherung (CLI)
- Hinzufügen von Tags zu vorhandenen Servern oder Sicherungen (CLI)
- Auflisten der Ressourcen-Tags (CLI)
- Löschen von Tags von einer Ressource (CLI)

Hinzufügen von Tags zu einem neuen OpsWorks for Puppet Enterprise Server (CLI)

Sie können den verwenden AWS CLI , um Tags hinzuzufügen, wenn Sie einen Server OpsWorks für Puppet Enterprise erstellen. In diesem Verfahren wird nicht vollständig beschrieben, wie ein Server erstellt wird. Ausführliche Informationen zum Erstellen eines Servers OpsWorks für Puppet Enterprise mithilfe des AWS CLI finden Sie Erstellen Sie einen Puppet Enterprise Master mit dem AWS CLI in diesem Handbuch. Sie können einem Server bis zu 50 Tags hinzufügen.

1. Stellen Sie sicher, dass alle Voraussetzungen für die Erstellung eines Servers OpsWorks für Puppet Enterprise erfüllt sind.
2. Führen Sie die Schritte 1-4 von Erstellen Sie einen Puppet Enterprise Master mit dem AWS CLI durch.

3. Fügen Sie beim Ausführen des `create-server`-Befehls in Schritt 5 den `--tags`-Parameter zu dem Befehl hinzu, wie im folgenden Beispiel gezeigt.

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

   Im Folgenden finden Sie ein Beispiel, das nur den Tag-Teil des `create-server`-Befehls zeigt.

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

4. Führen Sie die verbleibenden Schritte unter Erstellen Sie einen Puppet Enterprise Master mit dem AWS CLI aus. Führen Sie die Schritte unter Auflisten der Ressourcen-Tags (CLI) in diesem Thema aus, um zu überprüfen, ob Ihre Tags dem neuen Server hinzugefügt wurden.

Hinzufügen von Tags zu einer neuen Sicherung (CLI)

Sie können den verwenden AWS CLI , um Tags hinzuzufügen, wenn Sie ein neues, manuelles Backup eines Servers OpsWorks für Puppet Enterprise erstellen. In diesem Verfahren wird nicht vollständig beschrieben, wie eine manuelle Sicherung erstellt wird. Ausführliche Informationen zum Erstellen eines manuellen Backups finden Sie unter „So führen Sie ein manuelles Backup durch AWS CLI“ im "inEinen OpsWorks für Puppet Enterprise Server sichern. Sie können einer Sicherung bis zu 50 Tags hinzufügen. Wenn ein Server über Tags verfügt, werden neue Sicherungen automatisch mit den Tags des Servers markiert.

Wenn Sie einen neuen Server OpsWorks für Puppet Enterprise erstellen, sind automatische Backups standardmäßig aktiviert. Sie können Tags zu einer automatisierten Sicherung hinzufügen, indem Sie den unter Hinzufügen von Tags zu vorhandenen Servern oder Sicherungen (CLI) in diesem Thema beschriebenen `tag-resource`-Befehl ausführen.

- Führen Sie den folgenden Befehl aus, um einer manuellen Sicherung während der Erstellung der Sicherung Tags hinzuzufügen. Nur der Tag-Teil des Befehls wird angezeigt. Ein Beispiel für den vollständigen `create-backup`-Befehl finden Sie unter „So führen Sie eine manuelle Sicherung in der AWS CLI aus" in Einen OpsWorks für Puppet Enterprise Server sichern.

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

Das folgende Beispiel zeigt nur den Tag-Teil des `create-backup`-Befehls.

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

Hinzufügen von Tags zu vorhandenen Servern oder Sicherungen (CLI)

Sie können den `tag-resource` Befehl ausführen, um Tags zu vorhandenen Servern oder Backups OpsWorks für Puppet Enterprise hinzuzufügen (unabhängig davon, ob die Backups automatisch oder manuell erstellt wurden). Geben Sie den Amazon-Ressourcennamen (ARN) einer Zielressource an, um ihr Tags hinzuzufügen.

1. So rufen Sie den ARN der Ressource ab, auf die Sie Tags anwenden möchten:

   - Führen Sie für einen Server `describe-servers --server-name` *`server_name`* aus. Die Ergebnisse des Befehls zeigen den Server-ARN an.
   - Führen Sie für eine Sicherung `describe-backups --backup-id` *`backup_ID`* aus. Die Ergebnisse des Befehls zeigen den ARN der Sicherung an. Sie können auch ausführen`describe-backups --server-name` *`server_name`*, um Informationen über alle Backups für einen bestimmten OpsWorks Puppet Enterprise-Server anzuzeigen.

   Das folgende Beispiel zeigt nur die `ServerArn` in den Ergebnissen eines `describe-servers --server-name opsworks-cm-test`-Befehls an. Der `ServerArn`-Wert wird einem `tag-resource`-Befehl hinzugefügt, um dem Server Tags hinzuzufügen.

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

2. Führen Sie den `tag-resource`-Befehl mit dem ARN aus, den Sie in Schritt 1 erhalten haben.

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

Im Folgenden wird ein Beispiel gezeigt.

```
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. Um zu überprüfen, ob Tags erfolgreich hinzugefügt wurden, fahren Sie mit der nächsten Prozedur, Auflisten der Ressourcen-Tags (CLI), fort.

Auflisten der Ressourcen-Tags (CLI)

Sie können den `list-tags-for-resource` Befehl ausführen, um die Tags anzuzeigen, an die Puppet Enterprise-Server oder Backups angehängt sind. OpsWorks Geben Sie den ARN einer Zielressource an, um deren Tags anzuzeigen.

1. So rufen Sie den ARN der Ressource ab, für die Sie Tags auflisten möchten:
   - Führen Sie für einen Server `describe-servers --server-name` *`server_name`* aus. Die Ergebnisse des Befehls zeigen den Server-ARN an.
   - Führen Sie für eine Sicherung `describe-backups --backup-id` *`backup_ID`* aus. Die Ergebnisse des Befehls zeigen den ARN der Sicherung an. Sie können auch ausführen`describe-backups --server-name` *`server_name`*, um Informationen über alle Backups für einen bestimmten OpsWorks Puppet Enterprise-Server anzuzeigen.
2. Führen Sie den `list-tags-for-resource`-Befehl mit dem ARN aus, den Sie in Schritt 1 erhalten haben.

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

Im Folgenden wird ein Beispiel gezeigt.

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

Wenn Tags auf der Ressource vorhanden sind, gibt der Befehl Ergebnisse wie die folgenden zurück.

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

Löschen von Tags von einer Ressource (CLI)

Sie können den `untag-resource` Befehl ausführen, um Tags von OpsWorks Puppet Enterprise-Servern oder Backups zu löschen. Wenn die Ressource gelöscht wird, werden auch die Tags auf der Ressource gelöscht. Geben Sie den Amazon-Ressourcennamen (ARN) einer Zielressource an, um Tags von ihr zu entfernen.

1. So rufen Sie den ARN der Ressource ab, von der Sie Tags entfernen möchten:

   - Führen Sie für einen Server `describe-servers --server-name` *`server_name`* aus. Die Ergebnisse des Befehls zeigen den Server-ARN an.
   - Führen Sie für eine Sicherung `describe-backups --backup-id` *`backup_ID`* aus. Die Ergebnisse des Befehls zeigen den ARN der Sicherung an. Sie können auch ausführen`describe-backups --server-name` *`server_name`*, um Informationen über alle Backups für einen bestimmten OpsWorks Puppet Enterprise-Server anzuzeigen.

2. Führen Sie den `untag-resource`-Befehl mit dem ARN aus, den Sie in Schritt 1 erhalten haben. Geben Sie nur die Tags an, die Sie löschen möchten.

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

In diesem Beispiel entfernt der `untag-resource`-Befehl nur den Tag mit dem Schlüssel `Stage` und dem Wert `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. Führen Sie die Schritte unter Auflisten der Ressourcen-Tags (CLI) in diesem Thema aus, um zu überprüfen, ob Tags erfolgreich gelöscht wurden.

Weitere Informationen finden Sie unter:

- Erstellen Sie einen Puppet Enterprise Master mit dem AWS CLI
- Einen OpsWorks für Puppet Enterprise Server sichern
- AWS-Strategien für das Tagging
- Steuern des Zugriffs auf AWS Ressourcen mithilfe von Ressourcen-Tags im AWS Identity and Access Management Benutzerhandbuch
- Verwenden von Kostenzuordnungs-Tags im AWS Billing and Cost Management - Benutzerhandbuch.
- CreateBackup in der AWS OpsWorks CM-API-Referenz
- CreateServer in der AWS OpsWorks CM-API-Referenz
- TagResource in der AWS OpsWorks CM-API-Referenz
- ListTagsForResource in der AWS OpsWorks CM-API-Referenz
- UntagResource in der AWS OpsWorks CM-API-Referenz

Einen OpsWorks for Puppet Enterprise Server sichern und wiederherstellen

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere

Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Abschnitt wird beschrieben, wie Sie einen OpsWorks for Puppet Enterprise-Server sichern und wiederherstellen.

Themen

- Einen OpsWorks für Puppet Enterprise Server sichern
- Einen OpsWorks for Puppet Enterprise Server aus einem Backup wiederherstellen

Einen OpsWorks für Puppet Enterprise Server sichern

Important

Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können ein täglich oder wöchentlich wiederkehrendes Server-Backup OpsWorks für Puppet Enterprise definieren und den Service die Backups in Ihrem Namen in Amazon Simple Storage Service (Amazon S3) speichern lassen. Alternativ können Sie bei Bedarf manuelle Sicherungen durchführen.

Da Backups in Amazon S3 gespeichert werden, fallen zusätzliche Gebühren an. Sie können einen Aufbewahrungszeitraum für Backups von bis zu 30 Generationen definieren. Sie können über die AWS Supportkanäle eine Serviceanfrage stellen, um dieses Limit ändern zu lassen. Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Sie können den Backups eines Masters OpsWorks für Puppet Enterprise Tags hinzufügen. Wenn Sie einem Master OpsWorks für Puppet Enterprise Tags hinzugefügt haben, erben automatische Backups des Puppet Masters diese Tags. Weitere Informationen zum Hinzufügen und Verwalten

von Tags für Sicherungen finden Sie unter Mit Tags auf AWS OpsWorks for Puppet Enterprise Ressourcen arbeiten in diesem Handbuch.

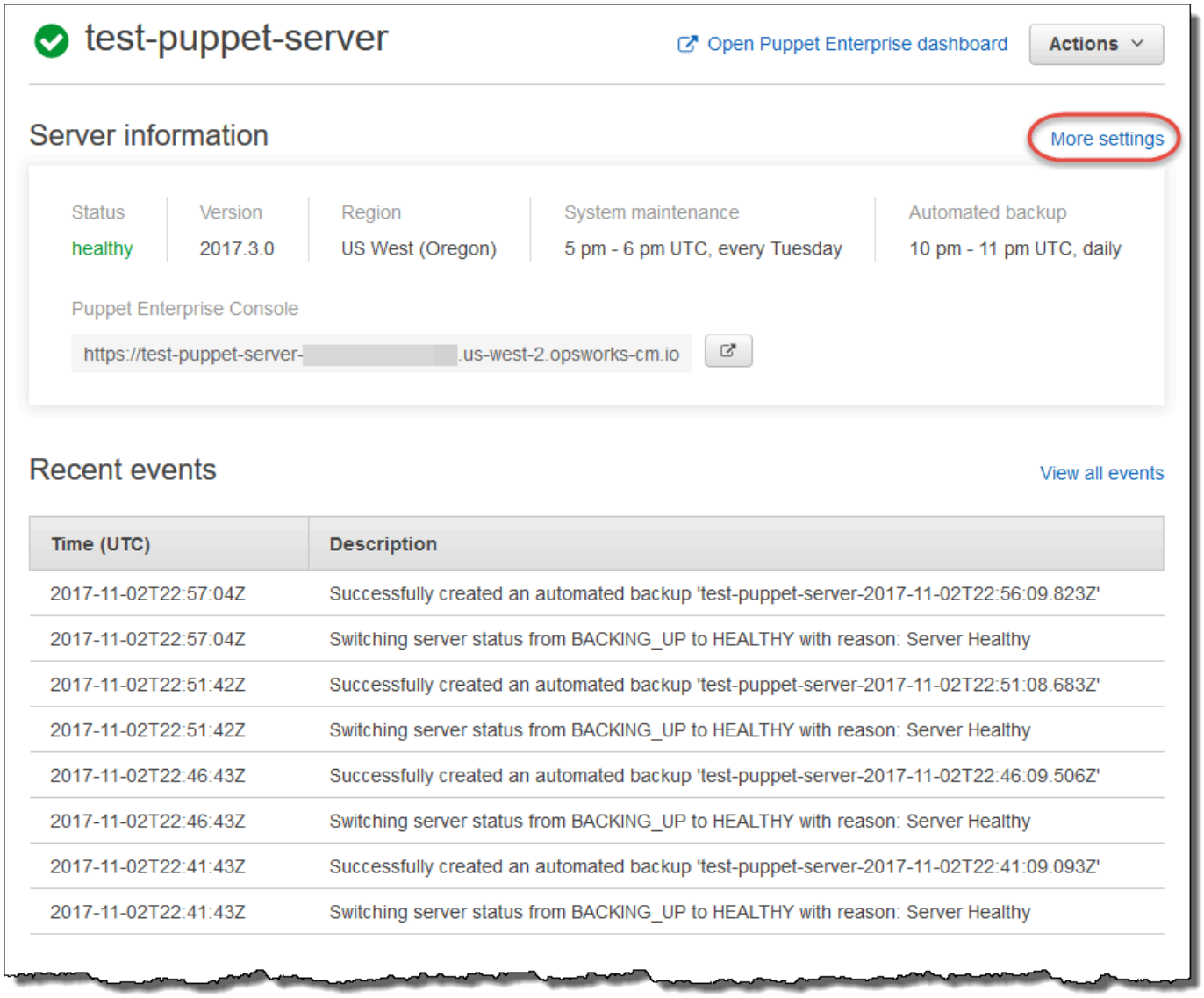
Themen

Automatische Backups

Wenn Sie Ihren Server OpsWorks für Puppet Enterprise konfigurieren, wählen Sie entweder automatische oder manuelle Backups. OpsWorks for Puppet Enterprise startet automatische Backups während der Stunde und an dem Tag, den Sie im Abschnitt Automatisches Backup auf der Seite Erweiterte Einstellungen konfigurieren des Setup-Assistenten ausgewählt haben. Nachdem Ihr Server online ist, können Sie die Sicherungseinstellungen ändern, indem Sie die folgenden Schritte auf der Eigenschaftenseite des Servers ausführen.

Ändern der Einstellungen für automatische Sicherungen

1. Wählen Sie auf der Eigenschaftenseite des Servers More settings (Weitere Einstellungen).



2. Um automatisierte Sicherungen zu deaktivieren, wählen Sie No (Nein) für die Option Enable automated backups (Automatische Sicherungen aktivieren) aus. Speichern Sie Ihre Änderungen. Sie müssen nicht zum nächsten Schritt gehen.
3. Ändern Sie im Abschnitt Automated Backup (Automatische Sicherung) die Häufigkeit, die Startzeit oder die Generationen, die aufbewahrt werden sollen. Speichern Sie Ihre Änderungen.

Manuelle Sicherungen

Sie können ein manuelles Backup jederzeit im oder starten AWS Management Console, indem Sie den Befehl AWS CLI create-backup ausführen. Manuelle Sicherungen sind nicht in den maximal 30 Generationen automatisierter Sicherungen enthalten, die gespeichert werden. Es werden maximal 10 manuelle Backups gespeichert, die manuell aus Amazon S3 gelöscht werden müssen.

Um ein manuelles Backup durchzuführen in der AWS Management Console

1. Wählen Sie auf der Seite Puppet Enterprise servers (Puppet Enterprise-Server) den Server aus, für den Sie eine Sicherung erstellen möchten.
2. Wählen Sie auf der Eigenschaftenseite im linken Navigationsbereich die Option Backups (Sicherungen).
3. Wählen Sie Create backup (Backup erstellen).
4. Die manuelle Sicherung ist fertig, wenn auf der Seite ein grünes Häkchen in der Spalte Status der Sicherung angezeigt wird.

Um ein manuelles Backup durchzuführen in der AWS CLI

Sie können Tags hinzufügen, wenn Sie ein neues, manuelles Backup eines Servers OpsWorks für Puppet Enterprise erstellen. Weitere Informationen zum Hinzufügen von Tags beim Erstellen einer manuellen Sicherung finden Sie unter Hinzufügen von Tags zu einer neuen Sicherung (CLI).

- Um ein manuelles Backup zu starten, führen Sie den folgenden AWS CLI Befehl aus.

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

Sicherungen löschen

Das Löschen einer Sicherung löscht diese endgültig aus dem S3-Bucket, in dem Sicherungen gespeichert werden.

Um ein Backup im zu löschen AWS Management Console

1. Wählen Sie auf der Seite Puppet Enterprise servers (Puppet Enterprise-Server) den Server aus, für den Sie eine Sicherung erstellen möchten.
2. Wählen Sie auf der Eigenschaftenseite im linken Navigationsbereich die Option Backups (Sicherungen).
3. Wählen Sie die Sicherung, die Sie löschen möchten, und wählen Sie dann Delete backup (Sicherung löschen). Sie können jeweils nur eine Sicherung auswählen.
4. Wenn Sie aufgefordert werden, das Löschen zu bestätigen, markieren Sie das Kontrollkästchen für Delete the backup, which is stored in an S3 bucket (Sicherung löschen, die in einem S3-Bucket gespeichert ist) und wählen Sie dann Yes, Delete (Ja, löschen).

Um ein Backup zu löschen in der AWS CLI

- Um ein Backup zu löschen, führen Sie den folgenden AWS CLI Befehl aus und ersetzen Sie den Wert von `--backup-id` durch die ID des Backups, das Sie löschen möchten. Backup-IDs haben das Format *`ServerName-yyyymmddhhmmssss`*. z. B. **`puppet-server-20171218132604388`**.

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

Einen OpsWorks for Puppet Enterprise Server aus einem Backup wiederherstellen

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Nachdem Sie Ihre verfügbaren Backups durchsucht haben, können Sie ganz einfach einen Zeitpunkt auswählen, ab dem Sie Ihren OpsWorks for Puppet Enterprise Server wiederherstellen möchten. Server-Sicherungen enthalten für die Konfigurationsverwaltungssoftware persistente Daten, wie Module, Klassen, Knotenzuordnungen, Datenbankinformationen (einschließlich von Berichten, Fakten usw.). Wenn Sie eine direkte Wiederherstellung eines Servers durchführen (d. h. den vorhandenen Server OpsWorks für Puppet Enterprise auf eine neue EC2-Instanz wiederherstellen), werden Knoten, die zum Zeitpunkt des Backups registriert waren, mit dem Sie den Server wiederherstellen, erneut registriert wurden, und der Datenverkehr wird auf die neue Instanz umgeleitet, wenn die Wiederherstellung erfolgreich ist und der Serverstatus OpsWorks für Puppet Enterprise wiederhergestellt ist`Healthy`. Bei der Wiederherstellung auf einem neu erstellten Server OpsWorks für Puppet Enterprise werden keine Knotenverbindungen aufrechterhalten. Beim Wiederherstellen eines Servers wird die Version der Puppet-Software nicht aktualisiert. Es gelten dieselbe Puppet-Version und dieselben Konfigurationsmanagement-Daten, die im Umfang der gewählten Sicherung verfügbar sind.

Das Wiederherstellen eines Servers nimmt in der Regel mehr Zeit in Anspruch als das Erstellen eines neuen Servers. Die Dauer hängt von der Größe des ausgewählten Backups ab. Nach Abschluss der Wiederherstellung bleibt die alte EC2-Instance im Zustand `Running` oder `Stopped`, jedoch nur vorübergehend. Dieser Zustand wird letztendlich beendet.

In dieser Version können Sie den verwenden, AWS CLI um einen Puppet-Master OpsWorks für Puppet Enterprise wiederherzustellen.

> **Note**
>
> Sie können auch den Befehl restore-server verwenden, um den aktuellen Instance-Typ zu ändern oder Ihren SSH-Schlüssel wiederherzustellen oder festzulegen, wenn er verloren ging oder beschädigt wurde.

Wiederherstellen eines Servers von einer Sicherung

1. Führen Sie in der den folgenden Befehl aus AWS CLI, um eine Liste der verfügbaren Backups und ihrer IDs zurückzugeben. Notieren Sie sich die ID der Sicherung, die Sie verwenden möchten. Backup-IDs haben das Format *myServerName-yyyymmddhhmmssss*.

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

2. Führen Sie den folgenden Befehl aus.

   ```
   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"
   ```

   Im Folgenden wird ein Beispiel gezeigt.

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

3. Warten Sie, bis die Wiederherstellung abgeschlossen ist.

Systemwartung OpsWorks für Puppet Enterprise

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Durch die obligatorische Systemwartung wird sichergestellt, dass die neuesten AWS getesteten Versionen von Puppet Server, einschließlich Sicherheitsupdates, immer auf einem Server OpsWorks für Puppet Enterprise ausgeführt werden. Die Systemwartung muss mindestens einmal pro Woche durchgeführt werden. Mithilfe von können Sie AWS CLI, falls gewünscht, die tägliche automatische Wartung konfigurieren. Sie können den auch verwenden AWS CLI , um neben der planmäßigen Systemwartung auch Systemwartungen bei Bedarf durchzuführen.

Wenn neue Versionen der Puppet-Software verfügbar werden, aktualisiert die Systemwartung die Version von Puppet Server automatisch auf dem Server, wenn sie den AWS-Test bestanden hat. AWS führt umfangreiche Tests durch, um sicherzustellen, dass Puppet-Upgrades produktionsbereit sind und bestehende Kundenumgebungen nicht stören. Daher kann es zu Verzögerungen zwischen Puppet-Softwareversionen und ihrer Verfügbarkeit für die Anwendung auf bestehenden Puppet Enterprise-Servern kommen. OpsWorks Weitere Informationen zum Aktualisieren verfügbarer Versionen von Puppet-Software auf Anfrage finden Sie Starten der Systemwartung nach Bedarf in diesem Thema.

Bei der Systemwartung wird eine neue Instance aus einem Backup gestartet, das im Rahmen des Wartungsprozesses durchgeführt wird. Dadurch wird das Risiko verringert, dass Amazon EC2 EC2-Instances, die regelmäßig gewartet werden, herabgesetzt oder beeinträchtigt werden.

> **Important**
>
> Bei der Systemwartung werden alle Dateien oder benutzerdefinierten Konfigurationen gelöscht, die Sie dem OpsWorks for Puppet Enterprise-Server hinzugefügt haben. Weitere Informationen zum Reparieren von Konfigurationen oder Wiederherstellen von Dateien

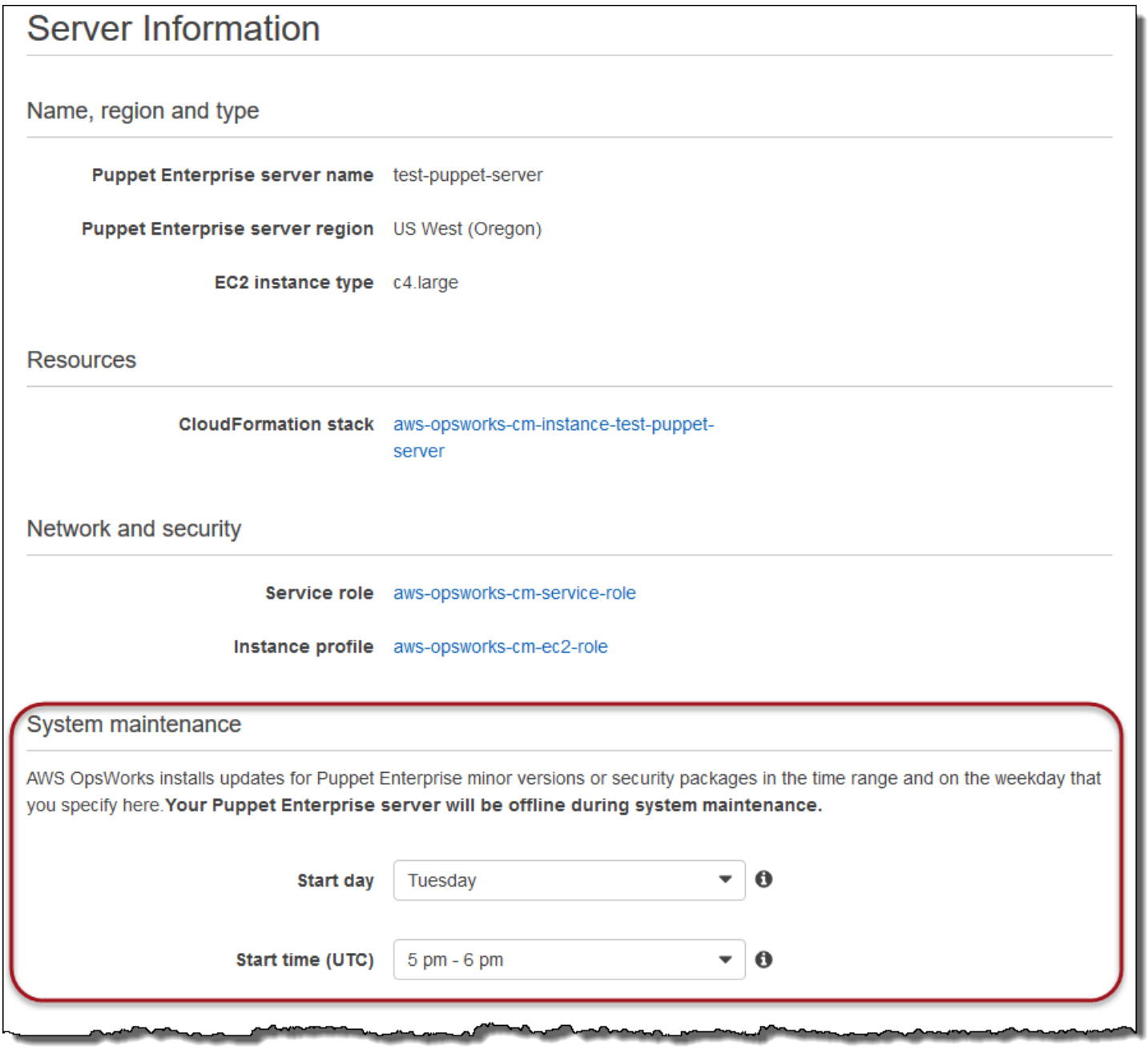
finden Sie unter Wiederherstellung benutzerdefinierter Konfigurationen und Dateien nach der Wartung in diesem Thema.

Themen

Konfigurieren der Systemwartung

Wenn Sie einen neuen Server OpsWorks für Puppet Enterprise erstellen, können Sie in koordinierter Weltzeit (UTC) einen Wochentag und eine Uhrzeit für den Beginn der Systemwartung konfigurieren. Die Wartung beginnt während der Stunde, die Sie angeben. Da der Server während der Systemwartung offline ist, wählen Sie eine Uhrzeit innerhalb der normalen Geschäftszeiten mit geringer Server-Nachfrage aus. Der Serverstatus ist `UNDER_MAINTENANCE`, während die Wartung läuft.

Sie können auch die Systemwartungseinstellungen auf einem vorhandenen Server OpsWorks für Puppet Enterprise ändern, indem Sie die Einstellungen im Bereich Systemwartung der Einstellungsseite für Ihren Server ändern, wie im folgenden Screenshot gezeigt.



Legen Sie im Abschnitt System maintenance (Systemwartung) den Tag und die Uhrzeit fest, zu der die Systemwartung beginnen soll.

Konfiguration der Systemwartung mit dem AWS CLI

Sie können die automatische Startzeit der Systemwartung auch mithilfe der AWS CLI konfigurieren. AWS CLI Damit können Sie bei Bedarf die tägliche automatische Wartung konfigurieren, indem Sie das dreistellige Wochentagspräfix weglassen.

Fügen Sie in einem `create-server`-Befehl den Parameter `--preferred-maintenance-window` Ihrem Befehl hinzu, nachdem Sie die Anforderungen zum Erstellen der Server-Instance angegeben haben (z. B. Instance-Typ, Instance-Profil-ARN und Service-Rollen-ARN). Im folgenden `create-server`-Beispiel ist `--preferred-maintenance-window` auf `Mon:08:00` eingestellt. Das bedeutet, dass Sie den Start der Wartung für jeden Montag um 08:00 Uhr festgelegt haben. festgelegt.

```
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 einem `update-server`-Befehl können Sie ggf. allein den Wert `--preferred-maintenance-window` aktualisieren. Im folgenden Beispiel wird das Wartungsfenster auf Freitag um 18:15 Uhr festgelegt. festgelegt.

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

Um den Beginn des Wartungsfensters auf jeden Tag um 18:15 Uhr (UTC) zu ändern, lassen Sie das aus drei Zeichen bestehende Präfix für den Wochentag weg, wie im folgenden Beispiel gezeigt.

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

Weitere Informationen zum Einstellen des bevorzugten Systemwartungsfensters mithilfe von finden Sie unter create-server AWS CLI und update-server.

Starten der Systemwartung nach Bedarf

Um die Systemwartung bei Bedarf außerhalb der konfigurierten wöchentlichen oder täglichen automatischen Wartung zu starten, führen Sie den folgenden Befehl aus. AWS CLI Sie können die Wartung nach Bedarf nicht in der AWS Management Console starten.

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

Weitere Informationen über diesen Befehl finden Sie unter start-maintenance.

Wiederherstellung benutzerdefinierter Konfigurationen und Dateien nach der Wartung

Bei der Systemwartung können benutzerdefinierte Dateien oder Konfigurationen, die Sie Ihrem OpsWorks for Puppet Enterprise-Server hinzugefügt haben, gelöscht oder geändert werden.

Wenn Ihrem Puppet-Master nach einem Wartungslauf Dateien oder Einstellungen fehlen, die Sie mithilfe von `RunCommand` oder SSH hinzugefügt haben, können Sie ein Amazon Machine Image (AMI) verwenden, um eine neue Amazon EC2 EC2-Instance zu starten. Es stehen AMIs zur Verfügung, die aus der Konfiguration eines Servers vor der Wartung erstellt wurden.

Die neue Instance befindet sich in demselben Zustand, in dem sich der Puppet-Master vor der Wartung befand, und sollte Ihre fehlenden Dateien und Einstellungen enthalten.

> **Important**
>
> Sie können die neue Instance nicht verwenden, um Ihren Server wiederherzustellen; die Instance kann nicht als Puppet-Master ausgeführt werden. Sie können die Instance nur verwenden, um Ihre Dateien und Konfigurationseinstellungen wiederherzustellen.

Um eine EC2-Instance von einem AMI aus zu starten, öffnen Sie in der Amazon EC2 EC2-Konsole den Startassistenten, wählen Sie Meine AMIs und dann das AMI aus, das Ihren Servernamen hat. Folgen Sie den Schritten des Amazon EC2 EC2-Assistenten wie bei jedem anderen Instance-Start.

Automatisches Hinzufügen von Knoten in OpsWorks Puppet Enterprise

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Thema wird beschrieben, wie Sie Ihrem OpsWorks for Puppet Enterprise-Server automatisch Amazon Elastic Compute Cloud (Amazon EC2) -Knoten hinzufügen. In Hinzufügen von Knoten, die vom Puppet-Master verwaltet werden, haben Sie erfahren, wie Sie mit dem Befehl `associate-node` einzelne Knoten nacheinander zu Ihrem Puppet Enterprise-Server hinzufügen. Der Code in diesem Thema veranschaulicht das automatische Hinzufügen von Knoten mit der unbeaufsichtigten Methode. Die empfohlene Methode für die unbeaufsichtigte (oder automatische) Zuordnung neuer Knoten ist die Konfiguration der Amazon EC2 EC2-Benutzerdaten. Standardmäßig ist ein Server OpsWorks für Puppet Enterprise bereits für die Node-Betriebssysteme Ubuntu, Amazon Linux und RHEL `puppet-agent`verfügbar.

Informationen zum Trennen der Zuordnung eines Knotens finden Sie Einen Knoten von einem OpsWorks for Puppet Enterprise Server trennen in diesem Handbuch und `disassociate-node`in der Dokumentation zur API OpsWorks für Puppet Enterprise.

Schritt 1: Erstellen Sie eine IAM-Rolle, die Sie als Ihr Instanzprofil verwenden können

Erstellen Sie eine AWS Identity and Access Management (IAM-) Rolle, die Sie als Ihr EC2-Instance-Profil verwenden möchten, und fügen Sie der IAM-Rolle die folgende Richtlinie hinzu. Diese Richtlinie ermöglicht der `opsworks-cm`-API die Kommunikation mit der EC2 Instance während der Knotenregistrierung. Weitere Informationen zu Instance-Profilen finden Sie unter Using Instance Profiles in der Amazon EC2 EC2-Dokumentation. Informationen zum Erstellen einer IAM-Rolle finden Sie unter Creating an IAM Role in the Console in der Amazon EC2 EC2-Dokumentation.

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

AWS OpsWorks stellt eine AWS CloudFormation Vorlage bereit, die Sie verwenden können, um die IAM-Rolle mit der vorherigen Richtlinienerklärung zu erstellen. Der folgende AWS CLI Befehl erstellt mithilfe dieser Vorlage die Instanzprofilrolle für Sie. Sie können den `--region` Parameter weglassen, wenn Sie den neuen AWS CloudFormation Stack in Ihrer Standardregion erstellen möchten.

```
aws cloudformation --region region ID create-stack --stack-name myPuppetinstanceprofile
 --template-url https://s3.amazonaws.com/opsworks-cm-us-east-1-prod-default-assets/
misc/owpe/opsworks-cm-nodes-roles.yaml --capabilities CAPABILITY_IAM
```

Schritt 2: Erstellen von Instances mit einem unbeaufsichtigten Skript für die Zuordnung

Um EC2-Instances zu erstellen, können Sie das Benutzerdatenskript, das im Starter Kit enthalten ist, in den `userdata` Abschnitt mit EC2-Instance-Anweisungen, Amazon EC2 Auto Scaling Scaling-Gruppenstartkonfigurationen oder in eine Vorlage kopieren. AWS CloudFormation Das Skript wird nur für EC2-Instances mit Ubuntu- und Amazon Linux-Betriebssystemen unterstützt. Weitere Informationen zum Hinzufügen von Skripten zu Benutzerdaten finden Sie unter Running Commands on Your Linux Instance at Launch in der Amazon EC2 EC2-Dokumentation. Der einfachste Weg, einen neuen Knoten zu erstellen, ist die Verwendung des Amazon EC2 EC2-Instance-Startassistenten. In dieser exemplarischen Vorgehensweise wird das unter beschriebene Beispiel-Setup des Apache-Webserver-Moduls verwendet. Erste Schritte mit OpsWorks für Puppet Enterprise

1. Das Benutzerdatenskript im Starter Kit führt den `opsworks-cm`-API-Befehl `associate-node` aus, um dem Puppet-Master einen neuen Knoten zuzuordnen. In dieser Version wird auch die aktuelle Version von AWS CLI auf dem Knoten für Sie installiert, falls dort nicht bereits die meiste up-to-date Version ausgeführt wird. Speichern Sie dieses Skript an einem leicht erreichbaren Speicherort als `userdata.sh`.

   Standardmäßig ist der Name des neu registrierten Knoten die Instance-ID.
2. Befolgen Sie die Anleitung in Starten einer Instance in der EC2-Dokumentation mit Änderungen. Wählen Sie im EC2-Instance-Startassistenten ein Amazon Linux AMI.
3. Wählen Sie auf der Seite Configure Instance Details (Instance-Details konfigurieren) myPuppetinstanceprofile, die Rolle, die Sie in Schritt 1: Erstellen Sie eine IAM-Rolle, die Sie als Ihr Instanzprofil verwenden können erstellt haben, als Ihre IAM-Rolle aus.

4. Laden Sie in den Bereich Advanced Details (Weitere Details) das `userdata.sh`-Skript hoch, das Sie in Schritt 1 erstellt haben.
5. Auf der Seite Add Storage (Speicher hinzufügen) sind keine Änderungen erforderlich. Gehen Sie weiter zu Add Tags (Tags hinzufügen).

   Durch die Anwendung von Tags auf Ihre EC2-Instance können Sie das Verhalten von `userdata.sh` anpassen. Für dieses Beispiel wenden Sie die Rolle `apache_webserver` auf Ihren Knoten an, indem Sie das folgende Tag hinzufügen: **`pp_role`** mit dem Wert **`apache_webserver`**.

   Wenn dem Knoten der Wert `pp_role` zugeordnet wird, werden Datenwerte festgelegt, die permanent im Agent-Zertifikat des Knoten gespeichert werden, sodass eine vertrauenswürdige Klassifizierung des Knoten möglich ist. Weitere Informationen finden Sie unter Extension-Anfragen (permanente Zertifikatdaten) in der Puppet-Plattform-Dokumentation.
6. Wählen Sie auf der Seite „Sicherheitsgruppe konfigurieren“ die Option Regel hinzufügen und wählen Sie dann den Typ HTTP aus, um in diesem Beispiel Port 8080 für den Apache-Webserver zu öffnen.
7. Wählen Sie Review and Launch (Überprüfen und starten) und dann Launch (Starten) aus. Wenn Ihr neuer Knoten gestartet wird, wendet er die Apache-Konfiguration des Beispielmoduls an, in Richten Sie das Starter Kit ein (Apache-Beispiel) dem Sie ihn eingerichtet haben.
8. Wenn Sie die Webseite öffnen, die mit dem öffentlichen DNS Ihres neuen Knotens verknüpft ist, sollten Sie eine Website sehen, die von Ihrem von Puppet verwalteten Apache-Webserver gehostet wird.

Einen Knoten von einem OpsWorks for Puppet Enterprise Server trennen

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Abschnitt wird beschrieben, wie Sie einen verwalteten Knoten von der Verwaltung durch einen OpsWorks for Puppet Enterprise-Server trennen oder entfernen können. Dieser Vorgang wird in der Befehlszeile oder in der Puppet Enterprise-Konsole ausgeführt. Sie können die Zuordnung von Knoten in der Managementkonsole von OpsWorks for Puppet Enterprise nicht trennen. Derzeit erlaubt die API OpsWorks für Puppet Enterprise nicht, mehrere Knoten stapelweise zu entfernen. Mit dem in diesem Abschnitt verwendeten Befehl wird ein Knoten nach dem anderen getrennt.

Es empfiehlt sich, Knoten von einem Puppet-Master zu trennen, bevor Sie den Server löschen, damit die Knoten im weiteren Verlauf nicht ständig versuchen, sich erneut mit dem Server zu verbinden. Führen Sie dazu den `disassociate-node` AWS CLI Befehl aus. Um einen Knoten vollständig aus PE zu entfernen, müssen Sie die Zuordnung des Knotens aufheben und sein Zertifikat widerrufen, damit der Knoten nicht ständig versucht, sich beim Puppet-Master anzumelden. Sie sollten außerdem `puppet-agent` von Knoten deinstallieren, wenn Sie sie nicht mehr im Puppet-Master verwalten wollen.

So heben Sie die Zuordnung von Knoten auf

1. Führen Sie in der den folgenden Befehl aus AWS CLI, um die Zuordnung von Knoten zu trennen. *`Node_name`* ist der Name des Knotens, dessen Zuordnung Sie aufheben möchten. Für Amazon EC2 EC2-Instances ist dies die Instance-ID. *`Server_name ist der Name`* des Puppet-Masters, von dem Sie die Zuordnung des Knotens trennen möchten. Beide Parameter sind erforderlich. Der Parameter `--region` wird nur benötigt, wenn Sie einen Knoten von einem Puppet-Master außerhalb der Standardregion trennen möchten.

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

   Nachfolgend finden Sie einen Beispielbefehl.

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

2. Warten Sie, bis in einer Antwortnachricht angezeigt wird, dass die Zuordnung aufgehoben wurde.

Weitere Hinweise zum Löschen eines OpsWorks für Puppet Enterprise bestimmten Servers finden Sie unter. Löschen Sie einen OpsWorks für Puppet Enterprise Server

Weitere Informationen finden Sie unter:

- Entfernen von Knoten in der Puppet Enterprise-Dokumentation

Löschen Sie einen OpsWorks für Puppet Enterprise Server

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Abschnitt wird beschrieben, wie Sie einen OpsWorks for Puppet Enterprise-Server löschen. Durch Löschen eines Servers werden auch seine auf dem Server gespeicherten Module gelöscht. Unterstützende Ressourcen (Amazon Elastic Compute Cloud-Instanz, Amazon Elastic Block Store-Volume usw.) werden zusammen mit allen automatisierten Backups ebenfalls gelöscht.

Obwohl durch Löschen eines Servers die Knoten nicht gelöscht werden, werden sie nicht mehr vom gelöschten Server verwaltet und versuchen fortlaufend, erneut eine Verbindung herzustellen. Aus diesem Grund empfehlen wir, die Zuordnung verwalteter Knoten aufzuheben, bevor Sie einen Puppet-Master löschen. In dieser Version können Sie Knoten trennen, indem Sie einen AWS CLI Befehl ausführen.

Schritt 1: Aufheben der Zuordnung von verwalteten Knoten

Heben Sie die Zuordnung von Knoten zum Puppet-Master auf, bevor Sie den Server löschen, damit die Knoten nicht fortlaufend versuchen, erneut eine Verbindung mit dem Server herzustellen. Führen Sie dazu den `disassociate-node` AWS CLI Befehl aus.

So heben Sie die Zuordnung von Knoten auf

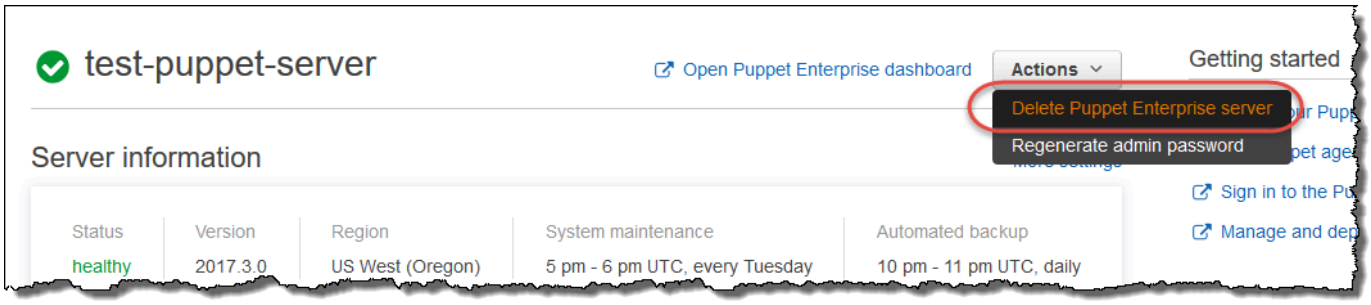
1. Führen Sie in der den folgenden Befehl aus AWS CLI, um die Zuordnung von Knoten zu trennen. *`Server_name`* ist der Name des Puppet-Masters, von dem Sie den Knoten trennen möchten. Der Wert von `--node-name` kann eine Instance-ID sein.

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

2. Warten Sie, bis in einer Antwortnachricht angezeigt wird, dass die Zuordnung aufgehoben wurde.

Schritt 2: Löschen des Servers

1. Erweitern Sie auf dem Dashboard auf der Serverkachel das Menü Actions (Aktionen).



2. Wählen Sie Delete Puppet Enterprise Server (Puppet Enterprise-Server löschen) aus.
3. Wenn Sie aufgefordert werden, das Löschen zu bestätigen, markieren Sie das Kontrollkästchen, um die zugehörigen Rollen und Ressourcen zu löschen, und wählen Sie dann Yes, Delete (Ja, löschen).

Weitere Informationen finden Sie unter:

- Einen Knoten von einem OpsWorks for Puppet Enterprise Server trennen

So migrieren Sie einen OpsWorks for Puppet Enterprise-Server zu Amazon Elastic Compute Cloud (Amazon EC2)

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Service hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In den folgenden Anweisungen wird beschrieben, wie Sie vorhandene Puppet Enterprise-Server zu Amazon EC2 migrieren, falls Sie Puppet Enterprise weiterhin für Ihre Konfigurationsmanagement-Anforderungen außerhalb von verwenden möchten. OpsWorks

Themen

Schritt 1: Wenden Sie sich an Puppet, um eine Lizenz zu erwerben

Wenn Sie Ihre Server auf EC2 migrieren, wird die neue Instanz nicht mit einer Puppet-Lizenz geliefert. Folgen Sie den Anweisungen auf der Puppet-Website, um einen Lizenzschlüssel zu erwerben.

Schritt 2: Informieren Sie sich über Ihren Server OpsWorks für Puppet Enterprise

Suchen und speichern Sie die Werte für Ihren Server OpsWorks für Puppet Enterprise.

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/s3/.

   Kopieren Sie den Namen des vorhandenen Amazon S3 S3-Buckets für Ihren OpsWorks for Puppet Enterprise-Server. Der Bucket-Name hat das Format: `aws-opsworks-cm-`*`server-name`*`-`*`random-string`*
2. Führen Sie den aws opsworks-cm describe-servers Befehl aus, um die Konfiguration für Ihren Server OpsWorks für Puppet Enterprise abzurufen.

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

```
    --server-name server-name \
    --region region
```

Speichern Sie die Werte für `InstanceTypeKeyPair`,`SubnetIds`, `SecurityGroupIdsInstanceProfileArn`, und `Endpoint` aus der Antwort.

3. Verwenden Sie SSH, um eine Verbindung zum vorhandenen Server OpsWorks für Puppet Enterprise herzustellen. Sie können den Sitzungsmanager in der EC2-Konsole anstelle von SSH verwenden.

   Führen Sie den folgenden Befehl aus.

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

   Die Antwort enthält die Puppet Enterprise-Version (z. B. 2019.8.10). Speichern Sie diesen Wert.

Für den nächsten Schritt verwenden Sie SSH oder Session Manager.

Schritt 3: Erstellen Sie ein Backup Ihres OpsWorks for Puppet Enterprise Servers

1. Führen Sie die folgenden Befehle aus, um ein lokales Backup zu erstellen.

   ```
   mkdir /tmp/puppet-backup/
   sudo /opt/puppetlabs/bin/puppet-backup create --dir=/tmp/puppet-backup/
   ```

2. Führen Sie den folgenden Befehl aus, um den Namen für das Backup zu speichern.

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

3. Führen Sie den folgenden Befehl aus, um Ihr Backup in einen S3-Bucket hochzuladen. Ersetzen Sie *`S3-Bucket`* durch den Wert aus Schritt 1 in. Schritt 2: Informieren Sie sich über Ihren Server OpsWorks für Puppet Enterprise

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

   Speichern Sie die Werte `PUPPET_BACKUP` und. `S3_BUCKET` Sie werden diese Werte in die neue EC2-Instance importieren.

Sie können die SSH- oder Session Manager-Sitzung beenden.

Schritt 4: Starten Sie eine neue EC2-Instance

Starten Sie eine neue EC2-Instance von der EC2-Konsole unter https://console.aws.amazon.com/ec2/ aus und verwenden Sie dabei dieselbe Konfiguration wie der OpsWorks for Puppet Enterprise-Server.

Parametername	Wert
OS	Amazon Linux 2
Instance-Typ	Der `InstanceType` Wert aus Schritt 2 von. Schritt 2: Informieren Sie sich über Ihren Server OpsWorks für Puppet Enterprise
Schlüsselpaarname	Der `KeyPair` Wert aus Schritt 2 vonSchritt 2: Informieren Sie sich über Ihren Server OpsWorks für Puppet Enterprise.
VPC	Die VPC des `SubnetIds` aus Schritt 2 vonSchritt 2: Informieren Sie sich über Ihren Server OpsWorks für Puppet Enterprise.
Subnetz	Die `SubnetIds` aus Schritt 2 von. Schritt 2: Informieren Sie sich über Ihren Server OpsWorks für Puppet Enterprise
Wählen Sie eine bestehende Sicherheitsgruppe aus -> Allgemeine Sicherheitsgruppen	Das `SecurityGroupIds` aus Schritt 2 vonSchritt 2: Informieren Sie sich über Ihren Server OpsWorks für Puppet Enterprise.
Speicherung	Mindestens 120 GB.
IAM-Instanzprofil	Das `InstanceProfileArn` aus Schritt 2 von. Schritt 2: Informieren Sie sich über Ihren Server OpsWorks für Puppet Enterprise

Wenn Sie eine Elastic IP erstellen und an die neue Instance anhängen möchten, kopieren Sie die Instance-ID der neuen Instance und führen Sie die Schritte unter aus(Optional) Schritt 4.1: Elastic IP erstellen und anhängen.

(Optional) Schritt 4.1: Elastic IP erstellen und anhängen

Durch Verwenden einer Elastic IP-Adresse können Sie Ausfälle bei Instances oder Software maskieren. Weisen Sie dazu die Adresse einer anderen Instance in Ihrem Konto neu zu.

Um eine Elastic IP-Adresse zu erstellen und zuzuordnen

1. Melden Sie sich bei der Amazon EC2 EC2-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie Elastic IPs.
3. Wählen Sie Elastic-IP-Adresse zuweisen aus.
4. Wählen Sie auf der Seite Elastic IP address zuweisen die Option Allocate aus. Dadurch wird eine öffentliche IPv4-Adresse erstellt.
5. Kopieren Sie die zugewiesene IPv4-Adresse.
6. Wählen Sie unter Aktionen die Option Elastic IP-Adresse zuordnen aus.
7. Geben Sie zum Beispiel die Instance-ID für die neue Instance ein.
8. Wählen Sie Associate aus.

Schritt 5: Installieren Sie Puppet Enterprise auf der neuen EC2-Instanz

Verwenden Sie SSH, um eine Verbindung zur neuen EC2-Instanz herzustellen. Sie können den Session Manager in der EC2-Konsole anstelle von SSH verwenden.

```
# 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
```

Sie können Ihre SSH- oder Session Manager-Sitzung für den nächsten Schritt geöffnet lassen.

Schritt 6: Stellen Sie das Backup auf der neuen EC2-Instanz wieder her

```
# 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
```

Sie können auf die Puppet-Konsole für die wiederhergestellte EC2-Instance unter https://*Öffentliches IPv4* der Instanz zugreifen. Sie finden das öffentliche IPv4-DNS auf der Detailseite der Instance in der EC2-Konsole. Bei den Anmeldeinformationen handelt es sich um dieselben Anmeldeinformationen, die Sie für den Zugriff auf Ihren OpsWorks for Puppet Enterprise-Server verwenden.

Sie können Ihre SSH- oder Session Manager-Sitzung für den nächsten Schritt geöffnet lassen.

Schritt 7: Konfigurieren Sie Ihre Puppet-Lizenz

Folgen Sie den Schritten auf der Puppet-Website, um Ihre Lizenz zu konfigurieren.

Sie können Ihre SSH- oder Session Manager-Sitzung für den nächsten Schritt geöffnet lassen.

Schritt 8: Migrieren Sie Ihre Knoten

Es gibt zwei Arten von Domänen, die von den OpsWorks vier Puppet Enterprise-Servern unterstützt werden:

- BYODC (Bringen Sie Ihre eigene Domain und Ihr eigenes Zertifikat mit)
- OpsWorks Endpunkt

Schritt 8.1: Für BYODC (Bringen Sie Ihre eigene Domain und Ihr eigenes Zertifikat mit)

Für diese Knoten müssen Sie lediglich die benutzerdefinierte Domain in Ihrem DNS-Anbieter auf die öffentliche IPv4-DNS- oder öffentliche IPv4-Adresse der neuen EC2-Instance verweisen.

Schritt 8.2: Für den Endpunkt OpsWorks

Für einen OpsWorks Endpunkt empfiehlt die Puppet-Dokumentation, den Puppet-Agent auf dem Knoten zu deinstallieren und dann den Puppet-Agent mithilfe des neu wiederhergestellten Puppet Enterprise-Servers zu installieren.

> **Note**
>
> Puppet verfügt zwar nicht über ein automatisiertes Verfahren zum Verschieben eines Agentenknotens, aber es gibt einige Module, die Mitglieder der Puppet-Community auf der Puppet Forge-Website veröffentlicht haben, um eine automatisierte Knotenmigration durchzuführen. Zu diesen Modulen gehören das `pe_migrate`Modul und ein zweites Migrationsmodul eines anderen Autors. Die Module auf der Puppet Forge-Website werden von Puppet nicht unterstützt, OpsWorks sofern nicht ausdrücklich im Forge-Modul darauf hingewiesen wird. Wir empfehlen, mit diesen Modulen Vorsicht walten zu lassen und sie zu testen, bevor sie allgemein verwendet werden.

In den folgenden Abschnitten werden die Schritte zur Deinstallation und Neuinstallation von Puppet-Agenten auf Linux-Instances beschrieben.

Themen

- Schritt 8.2.1: Kopieren Sie das Deinstallationsprogramm vom Puppet-Server
- Schritt 8.2.2: Laden Sie das Deinstallationsprogramm herunter und führen Sie es auf einem Knoten aus
- Schritt 8.2.3: Installieren Sie den Puppet-Agent erneut auf einem Knoten

Schritt 8.2.1: Kopieren Sie das Deinstallationsprogramm vom Puppet-Server

Stellen Sie vor der Deinstallation des Agenten sicher, dass das IAM-Instanzprofil des Knotens S3-Berechtigungen bereitstellt. ReadOnly

Führen Sie den folgenden Befehl aus, um das Deinstallationsprogramm vom Puppet-Server in den S3-Bucket zu kopieren.

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

Nachdem Sie den Befehl ausgeführt haben, können Sie sich von der SSH- oder Session Manager-Sitzung des Puppet-Servers abmelden.

Schritt 8.2.2: Laden Sie das Deinstallationsprogramm herunter und führen Sie es auf einem Knoten aus

Verwenden Sie SSH, um eine Verbindung zum Knoten herzustellen. Sie können Session Manager in der EC2-Konsole anstelle von SSH verwenden, wenn es sich bei dem Knoten um eine EC2-Instanz handelt.

```
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
```

Sie können Ihre SSH- oder Session Manager-Sitzung für den nächsten Schritt geöffnet lassen.

Schritt 8.2.3: Installieren Sie den Puppet-Agent erneut auf einem Knoten

Gehen Sie wie folgt vor, um den Puppet-Agent auf einem Knoten neu zu installieren.

Themen

Schritt 8.2.3.1: Installieren Sie den Puppet-Agent mit der richtigen Konfiguration

Führen Sie den folgenden Befehl aus, um den Puppet-Agent zu installieren.

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

Sie können Ihre SSH- oder Session Manager-Sitzung für Schritt 8.2.2.3 geöffnet lassen.

Schritt 8.2.3.2: Akzeptieren Sie das Zertifikat in der Puppet-Konsole

1. Gehen Sie zur Konsole des Puppet-Servers unter. `https://Public_IPv4_DNS`
2. Wählen Sie Zertifikate und dann Unsignierte Zertifikate aus.
3. Wählen Sie Accept, um das Zertifikat des Puppet-Agenten zu signieren.

Schritt 8.2.3.3: Checken Sie den Knoten in den Puppet Enterprise-Server ein

Führen Sie den folgenden Befehl auf dem Knoten aus, um ihn in den Server einzuchecken.

```
puppet agent -t
```

Der Knoten sollte jetzt in der Konsole des Puppet-Servers sichtbar sein.

Schritt 9: Löschen Sie Ihren Server OpsWorks für Puppet Enterprise

Sie können entweder die OpsWorks Konsole oder AWS CLI zum Löschen Ihres OpsWorks for Puppet Enterprise-Servers verwenden.

Um Ihren Server mit der OpsWorks Konsole zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS OpsWorks Konsole unter https://console.aws.amazon.com/opsworks/.
2. Wählen Sie im Navigationsbereich Puppet Enterprise-Server aus.
3. Wählen Sie auf der Seite Puppet Enterprise-Server den Server aus, den Sie löschen möchten.
4. Wählen Sie unter Aktionen die Option Puppet Enterprise-Server löschen aus.

Um Ihren Server mit dem zu löschen AWS CLI

Führen Sie den folgenden Befehl aus.

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

Protokollierung OpsWorks von Puppet Enterprise API-Aufrufen mit AWS CloudTrail

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

OpsWorks for Puppet Enterprise ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in OpsWorks Puppet Enterprise ausgeführt wurden. CloudTrail erfasst alle API-Aufrufe OpsWorks für Puppet Enterprise als Ereignisse, einschließlich Aufrufe von der OpsWorks for Puppet Enterprise-Konsole und von Codeaufrufen an die APIs OpsWorks für Puppet Enterprise. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen OpsWorks für Puppet Enterprise. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, an OpsWorks die Puppet Enterprise gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

OpsWorks Informationen zu Puppet Enterprise finden Sie unter CloudTrail

CloudTrail ist für Ihr AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn in OpsWorks Puppet Enterprise eine Aktivität auftritt, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen in der CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse

in Ihrem AWS Konto ansehen, suchen und herunterladen. Weitere Informationen finden Sie unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich der Ereignisse OpsWorks für Puppet Enterprise, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser standardmäßig für alle Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- Übersicht zum Erstellen eines Trails
- CloudTrail Unterstützte Dienste und Integrationen
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Alle Aktionen von OpsWorks for Puppet Enterprise werden von der API-Referenz OpsWorks für Puppet Enterprise protokolliert CloudTrail und sind in dieser Dokumentation dokumentiert. Aufrufe der CreateServerDescribeServersAktionen, und generieren beispielsweise Einträge in den CloudTrail Protokolldateien. CreateBackup

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Gibt an, ob die Anforderung mit Root- oder IAM-Benutzer-Anmeldeinformationen ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter dem CloudTrail UserIdentity-Element.

Grundlegendes zu Puppet OpsWorks Enterprise-Protokolldateieinträgen

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen

oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag für die `CreateServer` Aktion OpsWorks for Puppet Enterprise.


```
{"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"
}
```

Problembehandlung OpsWorks für Puppet Enterprise

> **Important**
>
> Der AWS OpsWorks for Puppet Enterprise Dienst hat am 31. März 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Dieses Thema enthält einige häufig auftretende Probleme OpsWorks bei Puppet Enterprise sowie Lösungsvorschläge für diese Probleme.

Themen

Allgemeine Tipps zur Problembehebung

Sollte es Ihnen nicht möglich sein, einen Puppet-Master zu erstellen oder damit zu arbeiten, können Sie Fehlermeldungen oder Protokolle einsehen, die Ihnen helfen, den Fehler zu beheben. Die folgenden Aufgaben beschreiben allgemeine Ausgangspunkte bei der Fehlerbehebung eines Puppet-Master-Problems. Weitere Informationen zu bestimmten Fehlern und Lösungen finden Sie im Abschnitt Behebung bestimmter Fehler dieses Themas.

- Verwenden Sie die OpsWorks for Puppet Enterprise-Konsole, um Fehlermeldungen anzuzeigen, falls ein Puppet-Master nicht gestartet werden kann. Fehlermeldungen im Zusammenhang mit dem Starten und Ausführen des Servers werden auf der Eigenschaftenseite des Puppet-Masters oben angezeigt. Fehler können von OpsWorks Puppet Enterprise- oder Amazon EC2-Diensten herrühren, die zur Erstellung eines Puppet-Masters verwendet werden. AWS CloudFormation Auf der Eigenschaftenseite können Sie auch Ereignisse sehen, die auf einem laufenden Server auftreten und Fehlerereignismeldungen beinhalten können.
- Zur Lösung der EC2-Probleme können Sie eine Verbindung zu Ihrer Server-Instance über SSH herstellen und die Protokolle überprüfen. EC2-Instance-Protokolle werden im `/var/log/aws/`

`opsworks-cm`-Verzeichnis gespeichert. Diese Protokolle erfassen Befehlsausgaben, während bei OpsWorks Puppet Enterprise ein Puppet-Master gestartet wird.

Behebung bestimmter Fehler

Themen

Der Server befindet sich im Status „Verbindung verloren“

Problem: Der Status eines Servers wird als Verbindung unterbrochen angezeigt.

Ursache: Dies tritt am häufigsten auf, wenn eine Entität außerhalb von Änderungen an einem Server OpsWorks für Puppet Enterprise oder dessen unterstützenden Ressourcen AWS OpsWorks vornimmt. AWS OpsWorks kann keine Verbindung zu Puppet Enterprise-Servern herstellen, die sich im Status Verbindung verloren befinden, um Wartungsaufgaben wie das Erstellen von Backups, das Anwenden von Betriebssystem-Patches oder das Aktualisieren von Puppet zu erledigen. Infolgedessen fehlen auf Ihrem Server möglicherweise wichtige Updates, er ist anfällig für Sicherheitsprobleme oder er funktioniert aus anderen Gründen nicht wie erwartet.

Lösung: Führen Sie die folgenden Schritte aus, um die Serververbindung wiederherzustellen.

1. Stellen Sie sicher, dass Ihre Servicerolle über alle erforderlichen Berechtigungen verfügt.

    a. Wählen Sie auf der Seite Einstellungen für Ihren Server unter Netzwerk und Sicherheit den Link für die Servicerolle aus, die der Server verwendet. Dadurch wird die Servicerolle zur Anzeige in der IAM-Konsole geöffnet.

    b. Vergewissern Sie sich, dass auf der Registerkarte Berechtigungen der Eintrag in der Liste der Berechtigungsrichtlinien aufgeführt `AWSOpsWorksCMServiceRole` ist. Wenn sie nicht

aufgeführt ist, fügen Sie die `AWSOpsWorksCMServiceRole` verwaltete Richtlinie manuell zur Rolle hinzu.

c. Stellen Sie auf der Registerkarte Vertrauensbeziehungen sicher, dass die Servicerolle über eine Vertrauensrichtlinie verfügt, die darauf vertraut, dass der `opsworks-cm.amazonaws.com` Dienst Rollen in Ihrem Namen übernimmt. Weitere Informationen zur Verwendung von Vertrauensrichtlinien mit Rollen finden Sie unter Ändern einer Rolle (Konsole) oder im AWS Sicherheits-Blogbeitrag So verwenden Sie Vertrauensrichtlinien mit IAM-Rollen.

2. Stellen Sie sicher, dass Ihr Instanzprofil über alle erforderlichen Berechtigungen verfügt.

   a. Wählen Sie auf der Seite Einstellungen für Ihren Server unter Netzwerk und Sicherheit den Link für das Instanzprofil aus, das der Server verwendet. Dadurch wird das Instanzprofil zur Anzeige in der IAM-Konsole geöffnet.

   b. Stellen Sie auf der Registerkarte Berechtigungen sicher, dass `AmazonEC2RoleforSSM` beide in der Liste der Berechtigungsrichtlinien aufgeführt `AWSOpsWorksCMInstanceProfileRole` sind. Wenn eine oder beide nicht aufgeführt sind, fügen Sie diese verwalteten Richtlinien manuell zur Rolle hinzu.

   c. Stellen Sie auf der Registerkarte Vertrauensbeziehungen sicher, dass die Servicerolle über eine Vertrauensrichtlinie verfügt, die darauf vertraut, dass der `ec2.amazonaws.com` Dienst Rollen in Ihrem Namen übernimmt. Weitere Informationen zur Verwendung von Vertrauensrichtlinien mit Rollen finden Sie unter Ändern einer Rolle (Konsole) oder im AWS Sicherheits-Blogbeitrag So verwenden Sie Vertrauensrichtlinien mit IAM-Rollen.

3. Stellen Sie in der Amazon EC2 EC2-Konsole sicher, dass Sie sich in derselben Region wie die Region des OpsWorks for Puppet Enterprise-Servers befinden, und starten Sie dann die EC2-Instance neu, die Ihr Server verwendet.

   a. *Wählen Sie die EC2-Instance mit dem Namen Servername aus. aws-opsworks-cm-instance-*

   b. Wählen Sie im Menü Instanzstatus die Option Reboot instance aus.

   c. Warten Sie bis zu 15 Minuten, bis Ihr Server neu gestartet und vollständig online ist.

4. Stellen Sie in der Konsole OpsWorks für Puppet Enterprise auf der Seite mit den Serverdetails sicher, dass der Serverstatus jetzt fehlerfrei ist.

Wenn der Serverstatus nach Durchführung der vorherigen Schritte immer noch Verbindung verloren lautet, versuchen Sie es mit einer der folgenden Methoden.

- Ersetzen Sie den Server, indem Sie einen neuen Server erstellen und das Original löschen. Wenn Daten auf dem aktuellen Server für Sie wichtig sind, stellen Sie den Server anhand einer aktuellen Sicherung wieder her und überprüfen Sie, ob die Daten aktuell sind, bevor Sie den ursprünglichen Server löschen, der nicht mehr reagiert.
- Wenden Sie sich an den AWS Support.

Servererstellung schlägt mit der Nachricht "Die angefragte Konfiguration wird derzeit nicht unterstützt" fehl

Problem: Sie versuchen einen Puppet Enterprise-Server zu erstellen. Die Servererstellung schlägt jedoch mit einer Fehlermeldung wie "Die angefragte Konfiguration wird derzeit nicht unterstützt" fehl. Bitte überprüfen Sie die Dokumentation auf unterstützte Konfigurationen.

Ursache: Ein nicht unterstützter Instance-Typ könnte für den Puppet-Master angegeben worden sein. Wenn Sie einen Puppet-Server in einer VPC erstellen möchten, die über eine nicht standardmäßige Tenancy verfügt, z. B. eine für dedizierte Instances, müssen alle Instances innerhalb der angegebenen VPC auch einer dedizierten oder Host-Tenancy angehören. Da einige Instance-Typen, z. B. t2, nur mit einem Standard-Abonnement verfügbar sind, könnte der Puppet-Master-Instance-Typ möglicherweise nicht durch die angegebene VPC unterstützt werden. Außerdem schlägt die Servererstellung fehl.

Lösung: Wenn Sie eine VPC mit einer Nicht-Standard-Tenancy auswählen, nutzen Sie einen m4-Instance-Typ, der eine dedizierte Tenancy unterstützt.

Die Amazon EC2 EC2-Instance des Servers konnte nicht erstellt werden

Problem: Die Servererstellung schlug mit einer ähnlichen Fehlermeldung wie dieser fehl: "Die folgende Ressource(n) konnte(n) nicht erstellt werden: [EC2Instance]. Fehler beim Empfang von 1 Ressource-Signal innerhalb der angegebenen Dauer."

Ursache: Dies ist am wahrscheinlichsten, da die EC2-Instance keinen Zugriff auf das Netzwerk hat.

Lösung: Stellen Sie sicher, dass die Instance über einen ausgehenden Internetzugang verfügt und der AWS Service-Agent Befehle ausgeben kann. Stellen Sie sicher, dass Ihre VPC (eine VPC mit einem einzigen öffentlichen Subnetz) DNS resolution (DNS-Auflösung) aktiviert hat und Ihr Subnetz die Einstellung Auto-assign Public IP (Öffentliche IP-Adresse automatisch zuweisen) aktiviert hat.

Service-Rollen-Fehler verhindert die Servererstellung

Problem: Die Servererstellung schlägt fehl und es wird eine Fehlermeldung angezeigt, die besagt: „Nicht autorisiert, sts auszuführen:AssumeRole.“

Ursache: Dies kann auftreten, wenn die Service-Rolle, die Sie nutzen, nicht über die erforderlichen Berechtigungen zum Erstellen eines neuen Servers verfügt.

Lösung: Öffnen Sie die OpsWorks for Puppet Enterprise-Konsole. Verwenden Sie die Konsole, um eine neue Servicerolle und eine Instanzprofilrolle zu generieren. Wenn Sie lieber Ihre eigene Servicerolle verwenden möchten, fügen Sie die AWSOpsWorksCMServiceRoleRichtlinie der Rolle hinzu. Vergewissern Sie sich, dass opsworks-cm.amazonaws.com unter den Diensten in den Vertrauensbeziehungen der Rolle aufgeführt ist. Stellen Sie sicher, dass der Servicerolle, die dem Puppet-Master zugeordnet ist, die verwaltete Richtlinie angehängt ist. AWSOpsWorksCMServiceRole

Elastisches Limit der IP-Adresse überschritten

Problem: Servererstellung schlägt fehl mit folgender Fehlermeldung: "Die folgende Ressource(n) konnte(n) nicht erstellt werden: [EIP, EC2Instance]. Ressourcenerstellung abgebrochen, die maximale Anzahl an Adressen wurde erreicht."

Ursache: Dieses Problem tritt auf, wenn Ihr Konto die maximale Anzahl an Elastic IP (EIP)-Adressen genutzt hat. Das standardmäßige EIP-Adressenlimit ist fünf.

Lösung: Sie können entweder bestehende EIP-Adressen freigeben oder solche löschen, die Ihr Konto nicht aktiv verwendet, oder Sie können sich an den AWS Kundensupport wenden, um das Limit an EIP-Adressen zu erhöhen, das mit Ihrem Konto verknüpft ist.

Unbeaufsichtigte Knotenzuordnung fehlgeschlagen

Problem: Die unbeaufsichtigte oder automatische Zuordnung neuer Amazon EC2 EC2-Knoten schlägt fehl. Knoten, die dem Puppet-Master hinzugefügt hätten werden sollen, werden nicht im Dashboard von Puppet Enterprise angezeigt.

Ursache: Dies kann auftreten, wenn Sie nicht über eine IAM-Rolle verfügen, die als Instance-Profil eingerichtet wurde und die gestattet, dass `opsworks-cm`-API-Aufrufe mit neuen EC2-Instances kommunizieren können.

Lösung: Fügen Sie Ihrem EC2-Instance-Profil eine Richtlinie an, die es erlaubt, dass die API-Aufrufe `AssociateNode` und `DescribeNodeAssociationStatus` mit EC2 zusammenarbeiten können, wie in Automatisches Hinzufügen von Knoten in OpsWorks Puppet Enterprise beschrieben.

Die Systemwartung schlägt fehl

AWS OpsWorks CM führt wöchentliche Systemwartungen durch, um sicherzustellen, dass die neuesten AWS getesteten Versionen von Puppet Server, einschließlich Sicherheitsupdates, immer auf einem Server OpsWorks für Puppet Enterprise ausgeführt werden. Wenn die Systemwartung aus irgendeinem Grund fehlschlägt, werden Sie über den Fehler AWS OpsWorks CM informiert. Weitere Hinweise zur Systemwartung finden Sie unterSystemwartung OpsWorks für Puppet Enterprise.

In diesem Abschnitt werden mögliche Fehlerursachen beschrieben und Lösungen vorgeschlagen.

Themen

- Ein Fehler im Servicerollen- oder Instanzprofil verhindert die Systemwartung

Ein Fehler im Servicerollen- oder Instanzprofil verhindert die Systemwartung

Problem: Die Systemwartung schlägt fehl und es wird die Fehlermeldung „Nicht autorisiert, sts auszuführen:AssumeRole“ oder eine ähnliche Fehlermeldung zu den Berechtigungen angezeigt.

Ursache: Dieses Problem kann auftreten, wenn entweder die von Ihnen verwendete Servicerolle oder das Instanzprofil nicht über ausreichende Berechtigungen für die Durchführung der Systemwartung auf dem Server verfügt.

Lösung: Stellen Sie sicher, dass Ihre Servicerolle und Ihr Instanzprofil über alle erforderlichen Berechtigungen verfügen.

1. Stellen Sie sicher, dass Ihre Servicerolle über alle erforderlichen Berechtigungen verfügt.
    a. Wählen Sie auf der Seite Einstellungen für Ihren Server unter Netzwerk und Sicherheit den Link für die Servicerolle aus, die der Server verwendet. Dadurch wird die Servicerolle zur Anzeige in der IAM-Konsole geöffnet.
    b. Vergewissern Sie sich auf der Registerkarte „Berechtigungen“, dass sie der Servicerolle zugeordnet `AWSOpsWorksCMServiceRole` ist. Wenn sie nicht aufgeführt `AWSOpsWorksCMServiceRole` ist, fügen Sie diese Richtlinie der Rolle hinzu.
    c. Vergewissern Sie sich, dass opsworks-cm.amazonaws.com unter den Diensten in den Vertrauensbeziehungen der Rolle aufgeführt ist. Weitere Informationen zur Verwendung von Vertrauensrichtlinien mit Rollen finden Sie unter Rolle ändern (Konsole) oder im AWS Sicherheits-Blogbeitrag So verwenden Sie Vertrauensrichtlinien mit IAM-Rollen.
2. Stellen Sie sicher, dass Ihr Instanzprofil über alle erforderlichen Berechtigungen verfügt.

a. Wählen Sie auf der Seite Einstellungen für Ihren Server unter Netzwerk und Sicherheit den Link für das Instanzprofil aus, das der Server verwendet. Dadurch wird das Instanzprofil zur Anzeige in der IAM-Konsole geöffnet.

b. Stellen Sie auf der Registerkarte Berechtigungen sicher, dass `AmazonEC2RoleforSSM` beide in der Liste der Berechtigungsrichtlinien aufgeführt `AWSOpsWorksCMInstanceProfileRole` sind. Wenn eine oder beide nicht aufgeführt sind, fügen Sie diese verwalteten Richtlinien manuell zur Rolle hinzu.

c. Stellen Sie auf der Registerkarte Vertrauensbeziehungen sicher, dass die Servicerolle über eine Vertrauensrichtlinie verfügt, die darauf vertraut, dass der `ec2.amazonaws.com` Dienst Rollen in Ihrem Namen übernimmt. Weitere Informationen zur Verwendung von Vertrauensrichtlinien mit Rollen finden Sie unter Ändern einer Rolle (Konsole) oder im AWS Sicherheits-Blogbeitrag So verwenden Sie Vertrauensrichtlinien mit IAM-Rollen.

Weitere Hilfe und Support

Wenn Ihr spezifisches Problem in diesem Thema nicht beschrieben wird oder Sie die Vorschläge in diesem Thema ausprobiert und weiterhin Probleme haben, besuchen Sie die AWS OpsWorks -Foren.

Sie können auch das AWS Support-Center besuchen. Das AWS Support Center ist die zentrale Anlaufstelle für die Erstellung und Verwaltung von AWS Support-Fällen. Das AWS Support Center enthält auch Links zu anderen hilfreichen Ressourcen wie Foren, technischen FAQs, Servicestatus und AWS Trusted Advisor.

AWS OpsWorks für Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Lebensende erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

AWS OpsWorks for Chef Automate ermöglicht es Ihnen, einen Chef Automate-Server in AWS auszuführen. Sie können einen Chef-Server innerhalb von Minuten bereitstellen und dessen Betrieb, Backups, Wiederherstellungen und Software-Upgrades AWS OpsWorks for Chef Automate übernehmen lassen. AWS OpsWorks for Chef Automate gibt Ihnen die Möglichkeit, sich auf die wichtigsten Aufgaben des Konfigurationsmanagements zu konzentrieren, anstatt einen Chef-Server zu verwalten.

Ein Chef Automate-Server verwaltet die Konfiguration der Knoten in Ihrer Umgebung, indem er festlegt `chef-client`, welche Chef-Rezepte auf den Knoten ausgeführt werden sollen, Informationen über Knoten speichert und als zentrales Repository für Ihre Chef-Kochbücher dient. AWS OpsWorks for Chef Automate bietet Chef-Server, die Premium-Funktionen von Chef Automate enthalten: Chef Infra und Chef. InSpec

Ein AWS OpsWorks for Chef Automate Server läuft auf einer Amazon Elastic Compute Cloud-Instance. AWS OpsWorks for Chef Automate Server sind so konfiguriert, dass sie die neueste Version von Amazon Linux (Amazon Linux 2) ausführen. Weitere Informationen über die Änderungen in dieser Version von Chef Automate finden Sie unter Chef Automate – Versionshinweise. In der folgenden Tabelle werden die Chef-Komponenten beschrieben, die auf einem AWS OpsWorks for Chef Automate Server installiert sind.

Name der Komponente	Beschreibung	Auf dem AWS OpsWorks for Chef Automate Server installierte Version
Chef Automate	Chef Automate ist ein Software-Paket für	2.0

Name der Komponente	Beschreibung	Auf dem AWS OpsWorks for Chef Automate Server installierte Version
	Unternehmensserver, das einen automatisierten Workflow für kontinuierliche Bereitstellung und Einblicke zu verwalteten Knoten in einer Web-basierten Verwaltungskonsole bietet. Chef Automate bietet Infrastrukturautomatisierung durch Einbeziehung von Chef Infra, Sicherheits- und Compliance-Informationen und Durchsetzung durch die Einbeziehung von Chef InSpec sowie automatisierte Bereitstellung durch die Einbeziehung von Chef Habitat. Weitere Informationen zu Chef Automate finden Sie unter Chef Automate auf der Chef-Website.	

Name der Komponente	Beschreibung	Auf dem AWS OpsWorks for Chef Automate Server installierte Version
Chef Infra	Das bisher als Chef Server bezeichnete Chef Infra Server verwendet den Chef Infra Client (`chef-client`)-Agenten, um den verwalteten Knoten kontinuierlich Konfigurationen bereitzustellen, sodass ein Sollstatus aufrecht erhalten bleibt. Weitere Informationen zu Infra finden Sie unter Chef Infra auf der Chef-Website.	12.x

Name der Komponente	Beschreibung	Auf dem AWS OpsWorks for Chef Automate Server installierte Version
Chefkoch InSpec	Chef InSpec beschreibt Sicherheits- und Compliance-Regeln, die von Softwareingenieuren, Betriebs- und Sicherheitsingenieuren gemeinsam genutzt werden können. Compliance, Sicherheit und andere Richtlinienanforderungen bilden das Frameworks für automatisierte Tests für verwaltete Knoten, mit denen der `chef-client` -Agent die konsistente Durchsetzung von Standards sicherstellen kann. Weitere Informationen zu finden Sie InSpec unter Chef InSpec auf der Chef-Website.	3.9.0

Die unterstützte Mindestversion von `chef-client` auf mit einem AWS OpsWorks for Chef Automate -Server verknüpften Knoten ist 13x. Wir empfehlen, mindestens 14.10.9 oder die aktuellste, `chef-client` stabilste Version auszuführen.

Wenn neue Nebenversionen der Chef-Software verfügbar werden, aktualisiert die Systemwartung die Nebenversion von Chef Automate und Chef Server automatisch auf dem Server, wenn sie den AWS-Test bestanden hat. AWS führt umfangreiche Tests durch, um sicherzustellen, dass Chef-Upgrades produktionsbereit sind und bestehende Kundenumgebungen nicht stören. Daher kann es zu Verzögerungen zwischen den Chef-Softwareversionen und ihrer Verfügbarkeit für die Anwendung auf bestehenden OpsWorks Chef Automate-Servern kommen. Die Systemwartung führt auch ein Upgrade Ihres Servers auf die neueste Version von Amazon Linux durch.

Sie können jeden lokalen Computer oder jede EC2-Instance verbinden, auf der ein unterstütztes Betriebssystem ausgeführt wird und die Netzwerkzugriff auf einen Server hat. AWS OpsWorks for Chef Automate Eine Liste der unterstützten Betriebssysteme für die Knoten, die Sie verwalten möchten, finden Sie auf der Chef-Website. Die `chef-client`-Agent-Software ist auf Knoten installiert, die Sie mit einem Chef-Server verwalten möchten.

Themen

Regionalunterstützung für AWS OpsWorks for Chef Automate

Die folgenden regionalen Endpunkte unterstützen AWS OpsWorks for Chef Automate Server. AWS OpsWorks for Chef Automate erstellt Ressourcen, die Ihren Chef-Servern zugeordnet sind, wie Instanzprofile, Benutzer und Servicerollen, auf demselben regionalen Endpunkt wie Ihr Chef-Server. Ihr Chef-Server muss sich in einer VPC befinden. Sie können eine VPC verwenden, die Sie erstellt oder bereits zur Verfügung haben, oder Sie verwenden die Standard-VPC.

- Region USA Ost (Ohio)
- Region USA Ost (Nord-Virginia)
- Region US West (N. California)
- Region USA West (Oregon)
- Region Asien-Pazifik (Tokio)
- Region Asien-Pazifik (Singapur)
- Region Asien-Pazifik (Sydney)
- Region Europa (Frankfurt)
- Europe (Ireland) Region

AWS OpsWorks Häufig gestellte Fragen zum Ende des Lebenszyklus von Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Lebensende erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren.

Themen

Wie werden bestehende Benutzer von diesem Lebensende betroffen sein?

Bestehende Kunden sind bis zum 5. Mai 2024, dem Enddatum OpsWorks für Chef Automate, nicht betroffen. Nach dem Ende des Lebenszyklus können Kunden ihre Server nicht mehr über die OpsWorks Konsole oder API verwalten.

Was passiert mit meinen Servern, wenn ich nichts unternehme?

Ab dem 5. Mai 2024 können Sie Ihre Server nicht mehr über die OpsWorks Konsole oder API verwalten. Zu diesem Zeitpunkt werden wir alle laufenden Verwaltungsfunktionen für Ihre Server wie Backups oder Wartungsarbeiten einstellen. Um die Auswirkungen auf die Kunden zu begrenzen, lassen wir alle EC2-Instances laufen, die Chef Automate-Server sichern. Ihre Lizenzen sind jedoch nicht mehr gültig, da die Nutzung nicht mehr im Rahmen des Servicevertrags OpsWorks für Chef Automate mit Chef abgedeckt (oder in Rechnung gestellt) wird. Sie müssen sich an Chef wenden, um eine neue Lizenz zu erhalten. Wenn Sie Chef kontaktieren, teilen Sie ihnen unbedingt mit, dass Sie bereits OpsWorks Kunde von Chef Automate sind und von OpsWorks dort wechseln.

Zu welchen Alternativen kann ich wechseln?

AWS und Progress Chef empfehlen Ihnen, auf ihr neues Chef SaaS-Angebot zu migrieren, damit Sie weiterhin von einem vollständig verwalteten Chef Automate-Service profitieren können. Um mit Chef SaaS zu beginnen, können Sie sich an Chef wenden, um Unterlagen zur Einrichtung eines Chef SaaS-Kontos und zur Übertragung Ihrer Daten und Knoten zu erhalten.

Wenn Chef SaaS Ihre Anforderungen nicht erfüllt, weil Sie Chef Automate lieber auf EC2-Instances in AWS Konten ausführen möchten, die Sie kontrollieren, bietet Chef mehrere Optionen, darunter ein AWS Marketplace Bring Your Own License (BYOL) -Modell und Self-Hosting auf EC2. Sie können sich an Progress Chef wenden, um weitere Informationen zur Durchführung einer solchen Umstellung zu erhalten.

Akzeptiert der Service immer noch neue Kunden?

Nein. AWS OpsWorks denn Chef Automate akzeptiert keine neuen Kunden mehr.

Wird das Ende des Lebens alle AWS-Regionen gleichzeitig betreffen?

Ja. Die API und die Konsole werden das Ende ihrer Nutzungsdauer erreichen und ab dem 5. Mai 2024 insgesamt unbrauchbar sein. AWS-Regionen Informationen darüber, AWS-Regionen wo AWS OpsWorks Chef Automate verfügbar ist, finden Sie in der Liste der AWS regionalen Dienste.

Welches Maß an technischem Support ist verfügbar?

AWS wird bis zum Ende des Lebenszyklus weiterhin das gleiche Maß an Support OpsWorks für Chef Automate bieten, das Kunden heute haben. Wenn Sie Fragen oder Bedenken haben, können Sie das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support kontaktieren. Für Unterstützung bei der Umstellung empfehlen wir Kunden, sich an Progress Chef zu wenden.

Ich bin ein aktueller Kunde von OpsWorks Chef Automate und muss einen Server in einem Konto starten, das den Dienst zuvor nicht genutzt hat. Bin ich dazu in der Lage?

Im Allgemeinen nicht, es sei denn, es liegen außergewöhnliche Umstände vor. Wenn du eine besondere Situation hast, kontaktiere das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support mit den Einzelheiten und der Begründung dafür und wir werden deine Anfrage prüfen.

Wird es im nächsten Jahr wichtige Feature-Releases geben?

Nein. Da der Dienst das Ende seiner Nutzungsdauer erreicht, werden wir keine neuen Funktionen veröffentlichen. Wir werden jedoch weiterhin die Sicherheit verbessern und die Server bis zum Ende der Nutzungsdauer wie erwartet verwalten.

Aktualisieren Sie einen AWS OpsWorks for Chef Automate Server auf Chef Automate 2

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Lebensende erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

Voraussetzungen für das Upgraden auf Chef Automate 2

Bevor Sie beginnen, sollten Sie die neuen Funktionen kennen, die mit Chef Automate 2 hinzugefügt werden, und die Funktionen, die Chef Automate 2 nicht unterstützt. Informationen über die neuen und nicht unterstützten Funktionen in Chef Automate 2 finden Sie in der Dokumentation zu Chef Automate 2 auf der Chef-Website.

Ein Server, auf dem Chef Automate 1 läuft, muss nach dem 1. November 2019 mindestens einmal erfolgreich gewartet worden sein, um für ein Upgrade infrage zu kommen.

Wie bei allen Wartungsarbeiten auf Ihrem AWS OpsWorks for Chef Automate Server ist der Server während des Upgrades offline. Sie sollten während des Upgrade-Prozesses eine Ausfallzeit von bis zu drei Stunden einplanen.

Sie benötigen für die Chef Automate Dashboard-Website die Anmeldeinformationen für diesen Server. Wenn das Upgrade abgeschlossen ist, sollten Sie sich beim Chef Automate Dashboard anmelden und sich vergewissern, dass Ihre Knoten und Konfigurationsinformationen nicht verändert wurden.

> **Important**
>
> Wenn Sie bereit sind, Ihren AWS OpsWorks for Chef Automate Server auf Chef Automate 2 zu aktualisieren, verwenden Sie zum Upgrade nur die Anweisungen hier. Da viele Upgrade-Prozesse, wie z. B. die Erstellung von Backups, AWS OpsWorks for Chef Automate automatisiert werden, sollten Sie die Upgrade-Anweisungen auf der Chef-Website nicht befolgen.

Informationen zum Upgrade-Prozess

Während des Upgrade-Prozesses wird Ihr Server vor Beginn und nach Abschluss des Upgrades gesichert. Folgende Sicherungen werden erstellt:

- Ein Sicherung des Servers, wenn auf ihm noch Chef Automate 1 (Version 12.17.33) ausgeführt wird.
- Eine Sicherung des Servers, nachdem das Upgrade abgeschlossen ist und auf dem Server Chef Automate 2 (Version 2019-08) ausgeführt wird.

Der Upgrade-Prozess beendet die Amazon EC2 EC2-Instance, die der Server verwendet hat, als er Chef Automate 1 ausgeführt hat. Es wird eine neue Instance wird erstellt, um den Chef Automate 2-Server auszuführen.

Upgrade auf Chef Automate 2 (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS OpsWorks Konsole unter https://console.aws.amazon.com/opsworks/.
2. Wählen Sie im linken Navigationsbereich die Option AWS OpsWorks for Chef Automate aus.
3. Wählen Sie einen Server, um sich seine Eigenschaftenseite anzeigen zu lassen. Ein blaues Banner oben auf der Seite sollte anzeigen, ob der Server für ein Upgrade auf Chef Automate 2 infrage kommt.

   > **Note**
   >
   > Ein Server, auf dem Chef Automate 1 läuft, muss nach dem 1. November 2019 mindestens einmal erfolgreich gewartet worden sein, um für ein Upgrade infrage zu kommen.

4. Wenn der Server für ein Upgrade infrage kommt, wählen Sie Start upgrade (Upgrade starten) aus.
5. Das Upgrade kann bis zu drei Stunden dauern. Während des Upgrade-Prozesses wird auf der Eigenschaftenseite der Serverstatus als Under maintenance (Wartungszustand) angezeigt.
6. Nach Abschluss des Upgrades werden auf der Eigenschaftenseite die folgenden beiden Meldungen angezeigt: Successfully upgraded to Automate 2 (Erfolgreiches Upgrade auf Automate 2) und Maintenance completed successfully (Wartung erfolgreich abgeschlossen). Der Serverstatus sollte HEALTHY (FEHLERFREI) lauten.
7. Melden Sie sich mit Ihren vorhandenen Anmeldeinformationen beim Chef Automate Dashboard an, und vergewissern Sie sich, ob Ihre Knoten korrekt melden.

Upgrade auf Chef Automate 2 (CLI)

1. (Optional) Wenn Sie nicht sicher sind, welche Ihrer AWS OpsWorks for Chef Automate Server für ein Upgrade in Frage kommen, führen Sie den folgenden Befehl aus. Achten Sie darauf, den `--region` Parameter hinzuzufügen, wenn Sie AWS OpsWorks for Chef Automate Server in einer AWS-Region auflisten möchten, die sich von Ihrer Standard-AWS-Region unterscheidet.

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

Suchen Sie in den Ergebnissen nach dem a-Wert von `true` für das Attribut `CHEF_MAJOR_UPGRADE_AVAILABLE`. Dies zeigt an, dass der Server für ein Upgrade auf Chef Automate 2 infrage kommt. Notieren Sie sich die Namen der AWS OpsWorks for Chef Automate Server, die für ein Upgrade in Frage kommen.

2. Führen Sie den folgenden Befehl aus und ersetzen Sie *`server_name`* durch den Namen eines AWS OpsWorks for Chef Automate Servers. Um statt einer routinemäßigen Systemwartung das Upgrade auf Chef Automate 2 durchzuführen, fügen Sie wie im Befehl gezeigt das `CHEF_MAJOR_UPGRADE` Engine-Attribut hinzu. Fügen Sie den `--region`-Parameter hinzu, wenn sich der Zielserver nicht in Ihrer AWS-Standardregion befindet. Sie können nur einen Server pro Befehl upgraden.

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

Wenn der Server aus irgendeinem Grund AWS OpsWorks for Chef Automate nicht aktualisiert werden kann, führt dieser Befehl zu einer Validierungsausnahme.

3. Das Upgrade kann bis zu drei Stunden dauern. Sie können den Upgrade-Status regelmäßig überprüfen, indem Sie den folgenden Befehl ausführen.

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

Suchen Sie in den Ergebnissen nach dem `Status`-Wert. Ein `Status` von `UNDER_MAINTENANCE` bedeutet, dass das Upgrade noch im Gange ist. Ein erfolgreiches Upgrade gibt Meldungen ähnlich den folgenden zurück.

```
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
```

Wenn das Upgrade nicht erfolgreich war, AWS OpsWorks for Chef Automate wird Ihr Server automatisch auf Chef Automate 1 zurückgesetzt.

Wenn das Upgrade erfolgreich war, der Server aber nicht mehr so funktioniert wie vor dem Upgrade (z. B. wenn verwaltete Knoten nicht melden), können Sie den Server manuell

zurücksetzen. Informationen zum manuellen Zurücksetzen finden Sie unter Einen AWS OpsWorks for Chef Automate Server auf Chef Automate 1 (CLI) zurücksetzen.

Einen AWS OpsWorks for Chef Automate Server auf Chef Automate 1 (CLI) zurücksetzen

Wenn der Upgrade-Vorgang fehlschlägt, AWS OpsWorks for Chef Automate wird Ihr Server automatisch auf Chef Automate 1 zurückgesetzt. Wenn das Upgrade erfolgreich war, der Server jedoch nicht mehr so funktioniert wie vor dem Upgrade, können Sie Ihren AWS OpsWorks for Chef Automate Server manuell auf Chef Automate 1 zurücksetzen, indem Sie den verwenden AWS CLI.

1. Führen Sie den folgenden Befehl aus, um die `BackupId` der letzten Sicherung anzuzeigen, die auf Ihrem Server vor dem Upgrade-Versuch durchgeführt wurde. Fügen Sie den `--region`-Parameter hinzu, wenn sich Ihr Server in einer AWS-Region befindet, die sich von Ihrer AWS-Standardregion unterscheidet.

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

   Backup-IDs haben das Format *`ServerName-yyyymmddhhmmssss`*. Suchen Sie in den Ergebnissen nach den folgenden Eigenschaften von Chef Automate 1.

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

2. Führen Sie den folgenden Befehl aus, wobei Sie die in Schritt 1 zurückgegebene Sicherungs-ID als Wert von `--backup-id` verwenden.

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

   Je nach Menge der Daten, die Sie auf dem Server gespeichert haben, dauert die Wiederherstellung des Servers zwischen 20 Minuten und drei Stunden. Während des Wiederherstellungsvorgangs weist der Server den Status `RESTORING` auf. Dieser Status wird auf der Eigenschaftenseite des Servers im angezeigt und in den Ergebnissen des Befehls zurückgegeben AWS Management Console. describe-servers

3. Nach Abschluss der Wiederherstellung zeigt die Konsole die Meldung Restore completed successfully (Wiederherstellung erfolgreich abgeschlossen) an. Ihr AWS OpsWorks for Chef Automate Server ist online, und zwar genauso wie vor Beginn des Upgrade-Vorgangs.

Weitere Informationen finden Sie unter:

- Systemwartung in AWS OpsWorks for Chef Automate
- Einen AWS OpsWorks for Chef Automate Server aus einem Backup wiederherstellen
- DescribeServers in der AWS OpsWorks -API-Referenz
- StartMaintenance in der AWS OpsWorks -API-Referenz

Erste Schritte mit AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

AWS OpsWorks for Chef Automate ermöglicht es Ihnen, einen Chef Automate-Server in zu betreiben. AWS Sie können in etwa 15 Minuten einen Chef-Server bereitstellen.

AWS OpsWorks for Chef Automate Speichert ab dem 3. Mai 2021 einige Chef Automate-Serverattribute in AWS Secrets Manager. Weitere Informationen finden Sie unter Integration in AWS Secrets Manager.

Die folgende exemplarische Vorgehensweise hilft Ihnen bei der Erstellung Ihres ersten Chef-Servers in AWS OpsWorks for Chef Automate.

Voraussetzungen

Bevor Sie beginnen, müssen Sie die folgenden Voraussetzungen erfüllen.

Themen

- Einrichten einer VPC
- Voraussetzungen für die Verwendung einer benutzerdefinierten Domäne (optional)
- Einrichten eines EC2-Schlüsselpaares (optional)

Einrichten einer VPC

Ihr AWS OpsWorks for Chef Automate Server muss in einer Amazon Virtual Private Cloud betrieben werden. Sie können den Server zu einer vorhandenen VPC hinzufügen, die Standard-VPC verwenden oder eine neue VPC erstellen. Informationen zu Amazon VPC und zur Erstellung einer neuen VPC finden Sie im Amazon VPC Getting Started Guide.

Selbst erstellte oder vorhandene VPCs müssen folgende Einstellungen oder Eigenschaften aufweisen.

- Die VPC sollte über mindestens ein Subnetz verfügen.

  Wenn Ihr AWS OpsWorks for Chef Automate Server öffentlich zugänglich sein soll, machen Sie das Subnetz öffentlich und aktivieren Sie Auto-Assign Public IP.
- DNS resolution (DNS-Auflösung) muss aktiviert sein.
- Aktivieren Sie auf dem Subnetz Auto-assign public IP (Öffentliche IP automatisch zuweisen).

Wenn Sie nicht damit vertraut sind, VPCs zu erstellen oder Ihre Instances darin auszuführen, können Sie den folgenden AWS CLI Befehl ausführen, um eine VPC mit einem einzigen öffentlichen Subnetz zu erstellen, indem Sie eine AWS CloudFormation Vorlage verwenden, die AWS OpsWorks für Sie bereitgestellt wird. Wenn Sie lieber die verwenden möchten AWS Management Console, können Sie die Vorlage auch auf die Konsole hochladen. AWS CloudFormation

```
aws cloudformation create-stack --stack-name OpsWorksVPC --template-url https://
s3.amazonaws.com/opsworks-cm-us-east-1-prod-default-assets/misc/opsworks-cm-vpc.yaml
```

Voraussetzungen für die Verwendung einer benutzerdefinierten Domäne (optional)

Sie können Ihren Chef Automate-Server in Ihrer eigenen Domäne einrichten und einen öffentlichen Endpunkt in einer benutzerdefinierten Domäne angeben, der als Endpunkt Ihres Servers verwendet werden soll. Wenn Sie eine benutzerdefinierte Domäne verwenden, sind alle der folgenden Anforderungen erforderlich, wie in diesem Abschnitt ausführlich beschrieben.

Themen

Einrichten einer benutzerdefinierten Domäne

Um Ihren Chef Automate-Server in Ihrer eigenen benutzerdefinierten Domäne ausführen zu können, benötigen Sie einen öffentlichen Endpunkt eines Servers, z. B. `https://aws.my-company.com`. Wenn Sie eine benutzerdefinierte Domäne angeben, müssen Sie auch ein Zertifikat und einen privaten Schlüssel bereitstellen, wie in den vorherigen Abschnitten beschrieben.

Um auf den Server zuzugreifen, nachdem Sie ihn erstellt haben, fügen Sie einen CNAME-DNS-Eintrag in Ihrem bevorzugten DNS-Dienst hinzu. Dieser Datensatz muss die benutzerdefinierte Domäne auf den Endpunkt (den Wert des Serverattributs `Endpoint`) verweisen, der durch den Erstellungsprozess des Chef Automate-Servers generiert wird. Sie können nicht mit dem generierten `Endpoint`-Wert auf den Server zugreifen, wenn der Server eine benutzerdefinierte Domain verwendet.

Anfordern eines Zertifikats

Zum Einrichten Ihres Chef Automate-Servers in Ihrer eigenen benutzerdefinierten Domäne benötigen Sie ein PEM-formatiertes HTTPS-Zertifikat. Dabei kann es sich um ein einzelnes, selbstsigniertes Zertifikat oder um eine Zertifikatkette handeln. Wenn Sie dieses Zertifikat angeben, müssen Sie auch eine benutzerdefinierte Domäne und einen privaten Schlüssel angeben, wenn Sie den Workflow Create Chef Automate Server (Chef Automate-Server erstellen) durchführen.

Für den Zertifikatwert gelten folgende Anforderungen:

- Sie können entweder ein selbstsigniertes, benutzerdefiniertes Zertifikat oder die vollständige Zertifikatkette bereitstellen.
- Das Zertifikat muss ein gültiges X509-Zertifikat oder eine Zertifikatkette im PEM-Format sein.
- Das Zertifikat muss zum Zeitpunkt des Hochladens gültig sein. Sie können ein Zertifikat nicht vor Beginn des Gültigkeitszeitraums (das Datum `NotBefore` des Zertifikats) oder nach Ablauf der Gültigkeit (das Datum `NotAfter` des Zertifikats) hochladen.
- Der allgemeine Name des Zertifikats oder die alternativen Antragstellernamen (SANs) des Zertifikats müssen, sofern vorhanden, mit dem benutzerdefinierten Domänenwert übereinstimmen.
- Das Zertifikat muss mit dem Wert des Felds Custom private key (Benutzerdefinierter privater Schlüssel) übereinstimmen.

Anfordern eines privaten Schlüssels

Um Ihren Chef Automate-Server in Ihrer eigenen benutzerdefinierten Domäne einzurichten, benötigen Sie einen privaten Schlüssel im PEM-Format, um eine Verbindung mit dem Server mithilfe von HTTPS herzustellen. Der private Schlüssel darf nicht verschlüsselt sein; er kann nicht durch ein Passwort oder eine Passphrase geschützt werden. Wenn Sie einen benutzerdefinierten privaten Schlüssel angeben, müssen Sie auch eine benutzerdefinierte Domäne und ein Zertifikat bereitstellen.

Einrichten eines EC2-Schlüsselpaares (optional)

Für die typische Verwaltung des Chef-Servers ist eine SSH-Verbindung weder erforderlich noch empfehlenswert. Sie können zur Durchführung der meisten Verwaltungsaufgaben auf Ihrem Chef-Server `knife`-Befehle verwenden.

Ein EC2-Schlüsselpaar ist erforderlich, um eine SSH-Verbindung mit dem Server herzustellen, wenn Sie das Passwort zum Anmelden beim Chef Automate-Dashboard vergessen haben oder ändern möchten. Sie können ein bestehendes Schlüsselpaar verwenden oder ein neues Schlüsselpaar erstellen. Weitere Informationen zum Erstellen eines neuen EC2-Schlüsselpaars finden Sie unter Amazon EC2 EC2-Schlüsselpaare.

Wenn Sie kein EC2-Schlüsselpaar benötigen, sind Sie bereit, einen Chef-Server zu erstellen.

Erstellen eines Chef Automate-Servers

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

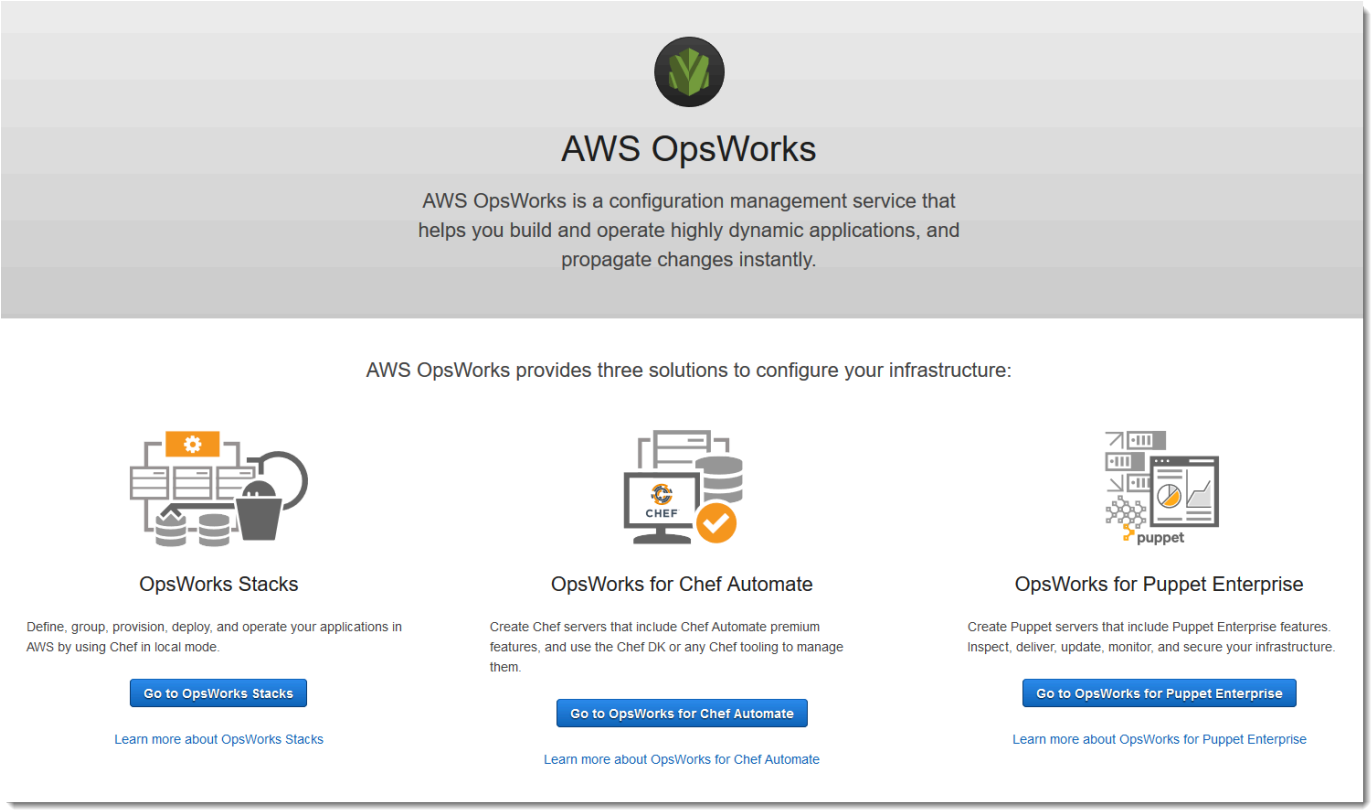
Sie können einen Chef-Server mithilfe der AWS OpsWorks for Chef Automate Konsole oder der erstellen. AWS CLI

Themen

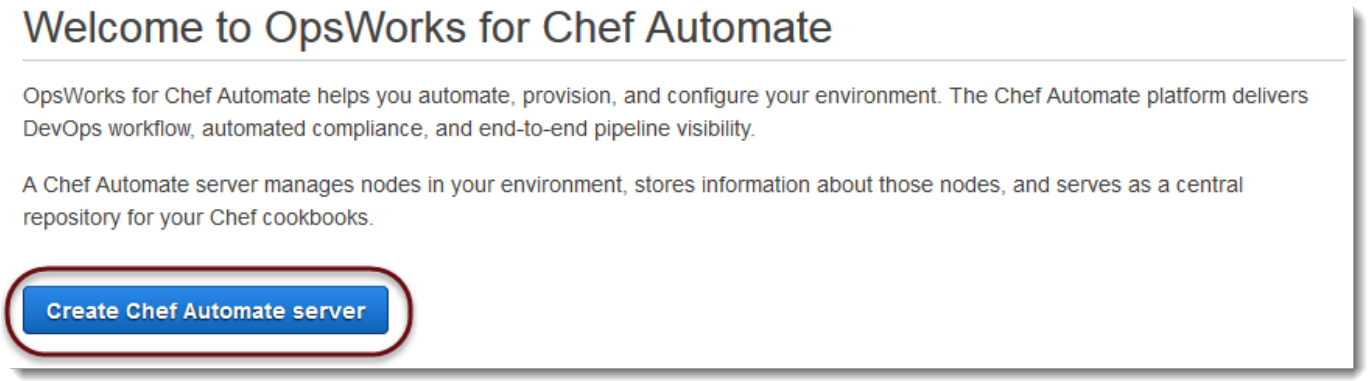
- Erstellen Sie einen Chef Automate-Server im AWS Management Console
- Erstellen Sie einen Chef Automate-Server mit dem AWS CLI

Erstellen Sie einen Chef Automate-Server im AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS OpsWorks Konsole unter https://console.aws.amazon.com/opsworks/.
2. Wählen Sie auf der AWS OpsWorks Startseite Go to OpsWorks für Chef Automate aus.

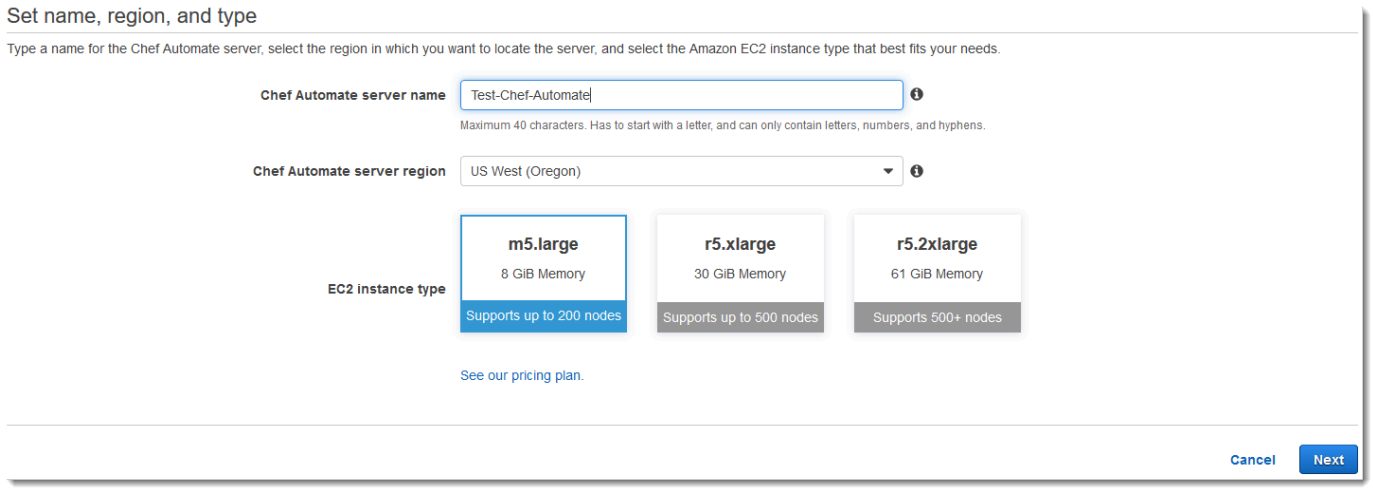


3. Wählen Sie auf der AWS OpsWorks for Chef Automate Startseite Create Chef Automate Server aus.

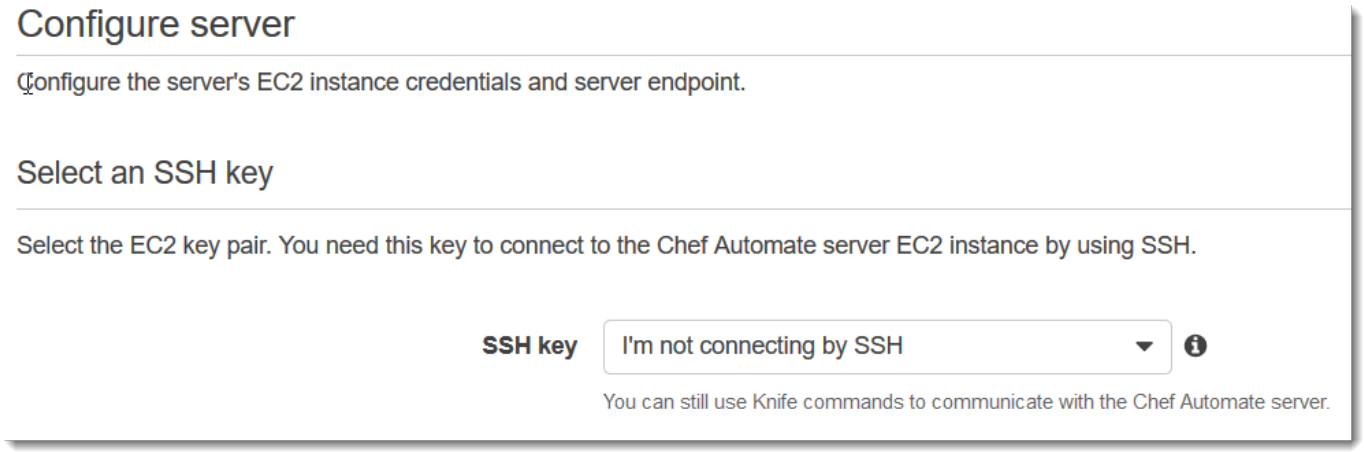


4. Geben Sie auf der Seite Set name, region, and type (Name, Region und Typ festlegen) einen Namen für Ihren Server an. Chef-Servernamen dürfen maximal 40 Zeichen lang sein und nur alphanumerische Zeichen und Bindestriche enthalten. Wählen Sie erst eine unterstützte Region

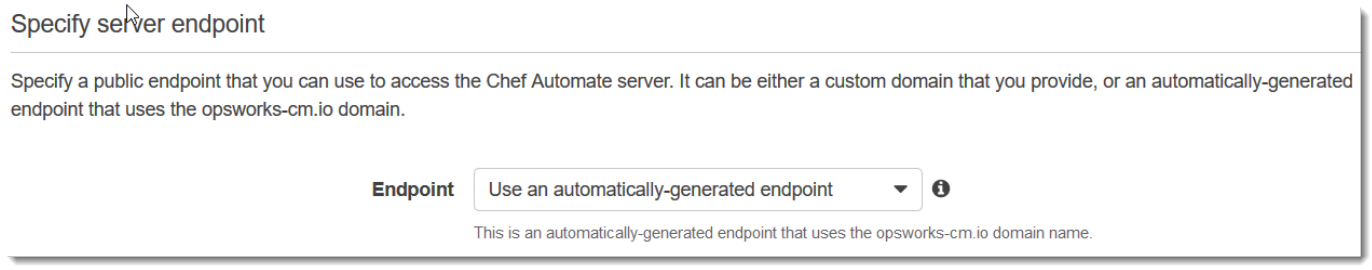
und dann einen Instance-Typ aus, der die Anzahl der Knoten unterstützt, die Sie verwalten möchten. Sie können bei Bedarf den Instance-Typ ändern, nachdem Ihr Server erstellt wurde. Für diese exemplarische Vorgehensweise erstellen wir einen Instance-Typ m5.large in der Region USA West (Oregon). Wählen Sie Weiter aus.



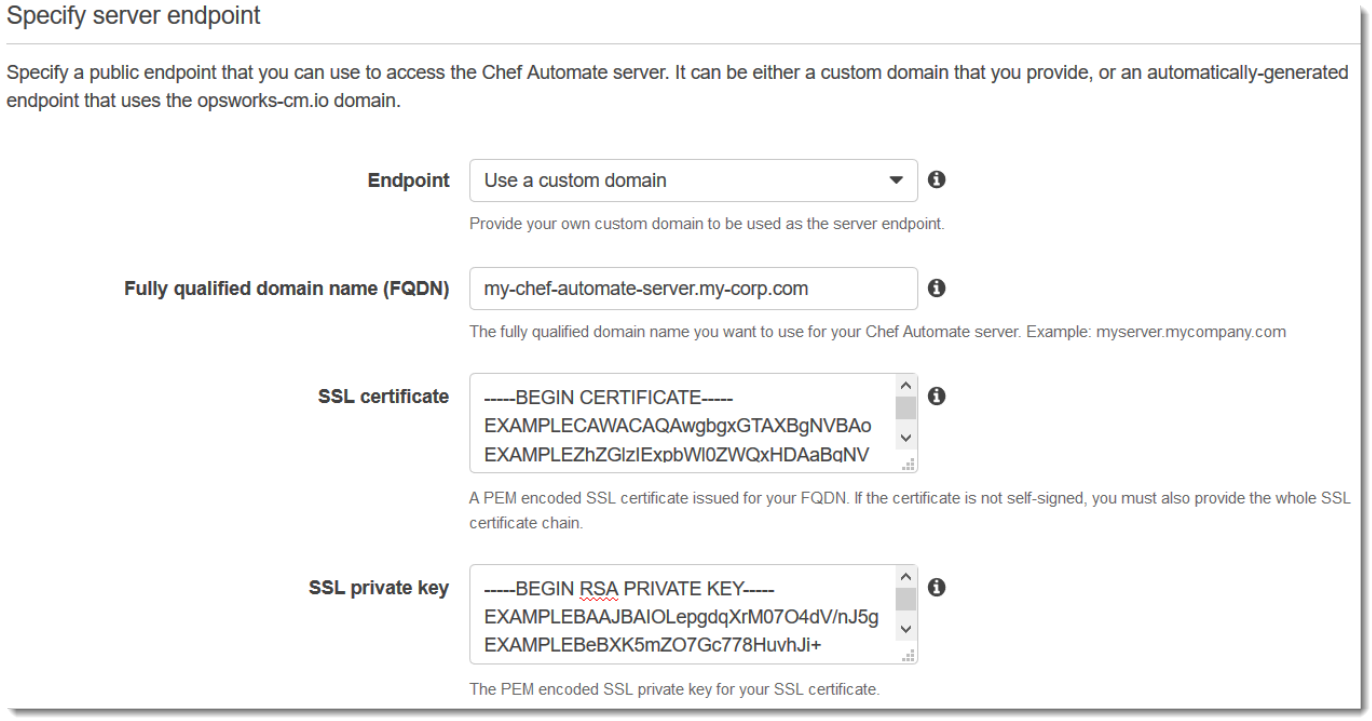
5. Behalten Sie auf der Seite Configure server (Server konfigurieren) in der Dropdown-Liste SSH key (SSH-Schlüssel) die Standardeinstellung bei, es sei denn, Sie möchten ein Schlüsselpaar angeben.



6. Behalten Sie unter Specify server endpoint (Serverendpunkt angeben) die Standardeinstellung Use an automatically generated endpoint (Automatisch generierten Endpunkt verwenden) bei und wählen Sie dann Next (Weiter), es sei denn, der Server soll sich in einer eigenen benutzerdefinierten Domäne befinden. Fahren Sie mit dem nächsten Schritt fort, um eine benutzerdefinierte Domäne zu konfigurieren.



7. Um eine benutzerdefinierte Domäne zu verwenden, wählen Sie unter Specify server endpoint (Serverendpunkt angeben) aus der Dropdown-Liste die Option Use a custom domain (Benutzerdefinierte Domäne verwenden) aus.



   a. Geben Sie für Fully Qualified Domain Name (FQDN) einen FQDN an. Sie müssen Eigentümer des Domänennamens sein, den Sie verwenden möchten.

   b. Fügen Sie unter SSL certificate (SSL-Zertifikat) das gesamte PEM-formatierte Zertifikat ein, beginnend mit `-----BEGIN CERTIFICATE-----` und endend mit `-----END CERTIFICATE-----`. Der Antragssteller des SSL-Zertifikats muss mit dem FQDN übereinstimmen, den Sie im vorherigen Schritt eingegeben haben.

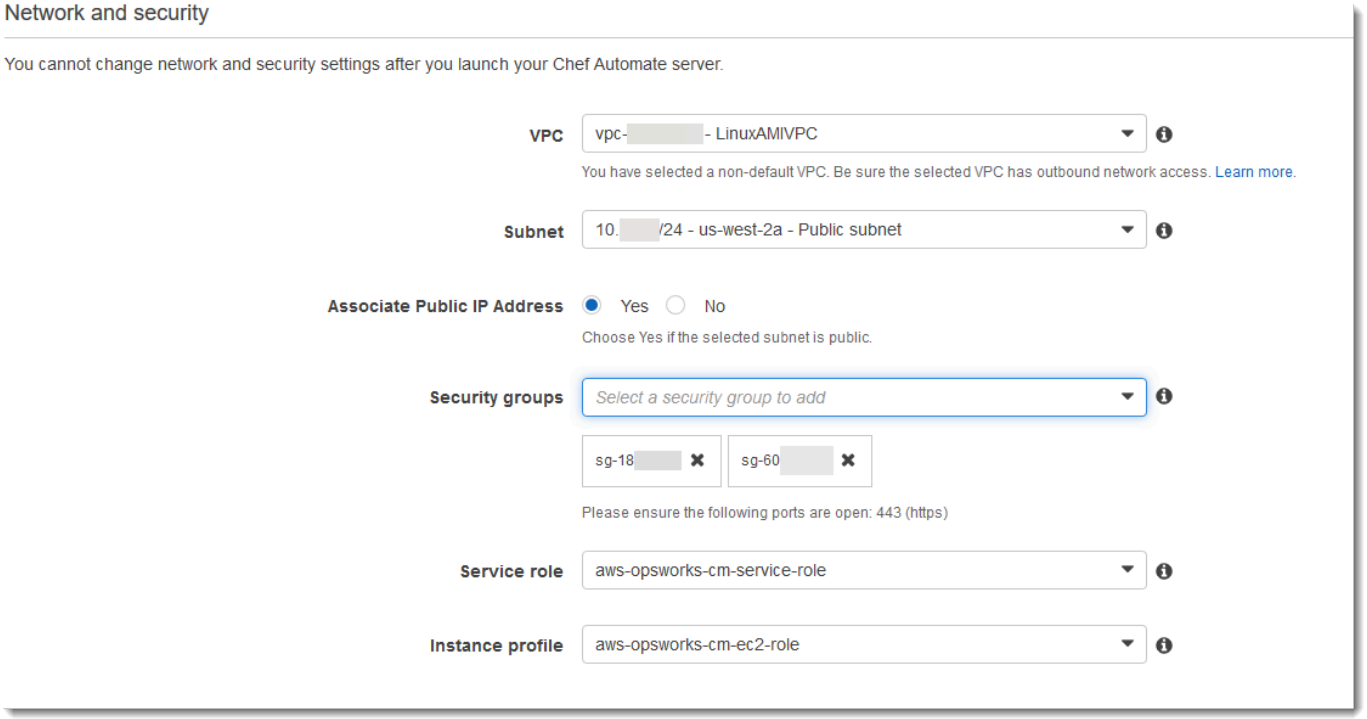
   c. Fügen Sie unter SSL private key (Privater SSL-Schlüssel) den gesamten privaten RSA-Schlüssel ein, beginnend mit `-----BEGIN RSA PRIVATE KEY-----` und endend mit `-----END RSA PRIVATE KEY-----`. Der private SSL-Schlüssel muss mit dem

öffentlichen Schlüssel im SSL-Zertifikat übereinstimmen, das Sie im vorherigen Schritt eingegeben haben. Wählen Sie Weiter aus.

8. Wählen Sie auf der Seite Configure advanced settings (Erweiterte Einstellungen konfigurieren) im Bereich Network and security (Netzwerk und Sicherheit) eine VPC, ein Subnetz und mindestens eine Sicherheitsgruppe aus. Die folgenden Anforderungen gelten für Ihre VPC:

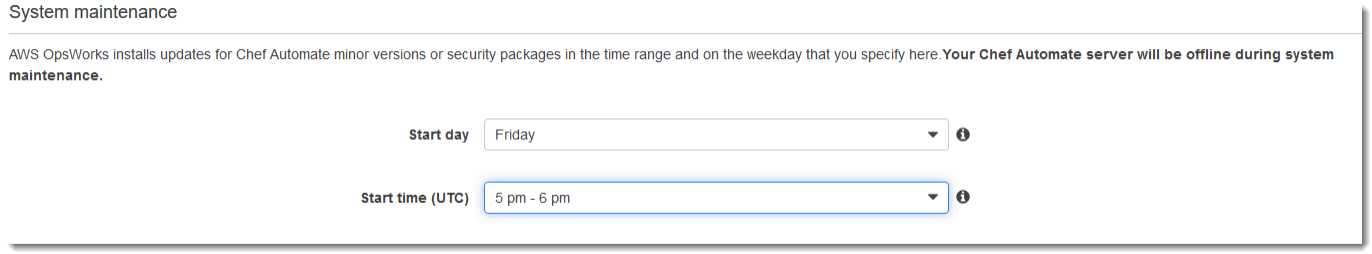
   - Die VPC muss über mindestens ein öffentliches Subnetz verfügen.
   - Die DNS-Auflösung muss aktiviert sein.
   - Auto-assign public IP (Öffentliche IP automatisch zuweisen) muss in öffentlichen Subnetzen aktiviert sein.

   AWS OpsWorks kann eine Sicherheitsgruppe, eine Servicerolle und ein Instanzprofil für Sie generieren, falls Sie noch keine haben, die Sie verwenden möchten. Der Server kann zu mehreren Sicherheitsgruppen gehören. Sie können die Netzwerk- und Sicherheitseinstellungen für den Chef-Server nicht ändern, nachdem Sie diese Seite verlassen haben.

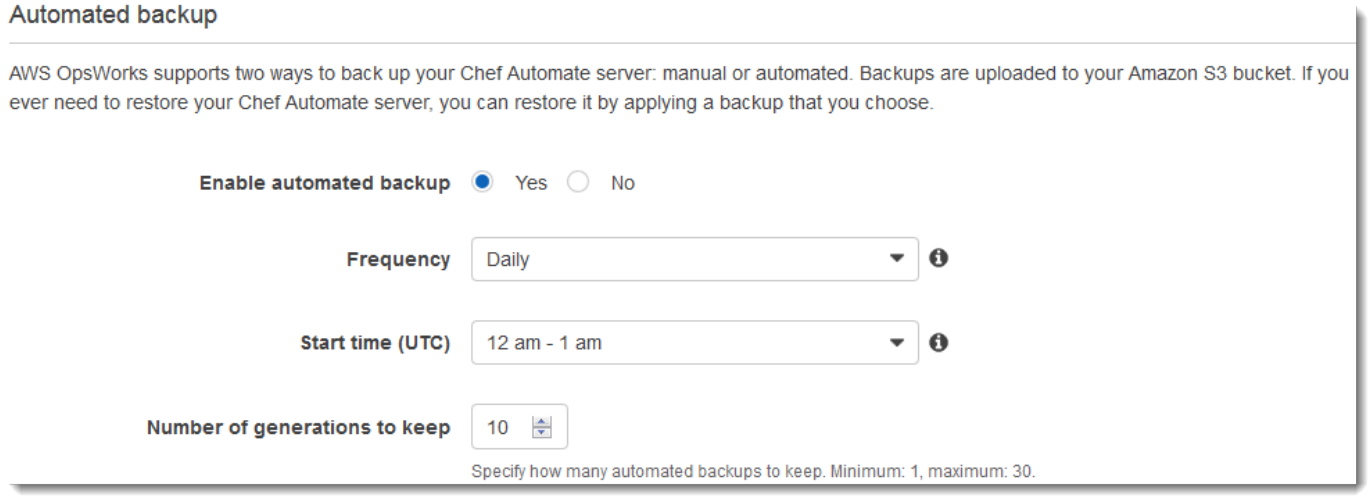


9. Legen Sie im Abschnitt System maintenance (Systemwartung) den Tag und die Uhrzeit fest, zu der die Systemwartung beginnen soll. Da der Server während der Systemwartung offline ist, wählen Sie eine Uhrzeit innerhalb der normalen Geschäftszeiten mit geringer Server-Nachfrage aus. Die verbundenen Knoten nehmen den Zustand `pending-server` an, bis die Wartung abgeschlossen ist.

Das Wartungsfenster muss angegeben werden. Sie können den Starttag und die Startzeit später ändern AWS Management Console, indem Sie die APIs AWS CLI, oder verwenden.



10. Konfigurieren Sie die Sicherungen. Standardmäßig sind automatische Sicherungen aktiviert. Legen Sie eine bevorzugte Häufigkeit und Stunde für den Start der automatischen Sicherung sowie die Anzahl der Backup-Generationen fest, die in Amazon Simple Storage Service gespeichert werden sollen. Es werden maximal 30 Backups aufbewahrt. Wenn das Maximum erreicht ist, werden die ältesten Backups AWS OpsWorks for Chef Automate gelöscht, um Platz für neue zu schaffen.



11. (Optional) Fügen Sie dem Server und den verwandten Ressourcen unter Tags Tags hinzu, z. B. die EC2-Instance, die Elastic IP-Adresse, die Sicherheitsgruppe, den S3-Bucket und Sicherungen. Weitere Hinweise zum Markieren eines AWS OpsWorks for Chef Automate Servers finden Sie unter. Mit Tags auf AWS OpsWorks for Chef Automate Ressourcen arbeiten
12. Wenn Sie die erweiterten Einstellungen fertig konfiguriert haben, wählen Sie Next (Weiter) aus.
13. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Wenn Sie bereit sind, den Server zu erstellen, wählen Sie Launch (Starten) aus.

    Während Sie darauf warten, Ihren Chef-Server AWS OpsWorks zu erstellen, fahren Sie fort Konfigurieren des Chef-Servers mit dem Starter Kit und laden Sie das Starter Kit und die

Anmeldeinformationen für das Chef Automate-Dashboard herunter. Warten Sie nicht mit dem Herunterladen dieser Elemente, bis Ihre Server online ist.

Nach Beendigung der Servererstellung ist Ihr Chef-Server auf der AWS OpsWorks for Chef Automate -Homepage mit dem Status online (Online) verfügbar. Nachdem sich der Server online befindet, ist das Chef Automate-Dashboard auf der Server-Domäne mit einer URL mit folgendem Format verfügbar: `https://your_server_name-random.region.opsworks-cm.io`.

Erstellen Sie einen Chef Automate-Server mit dem AWS CLI

Das Erstellen eines AWS OpsWorks for Chef Automate Servers durch Ausführen von AWS CLI Befehlen unterscheidet sich vom Erstellen eines Servers in der Konsole. AWS OpsWorks Erstellt in der Konsole eine Dienstrolle und eine Sicherheitsgruppe für Sie, falls Sie keine vorhandenen angeben, die Sie verwenden möchten. In der AWS OpsWorks kann eine Sicherheitsgruppe für Sie erstellen AWS CLI, wenn Sie keine angeben, aber es wird nicht automatisch eine Dienstrolle erstellt. Sie müssen einen Dienstrollen-ARN als Teil Ihres `create-server` Befehls angeben. Während AWS OpsWorks Sie Ihren Chef Automate-Server erstellen, laden Sie in der Konsole das Chef Automate-Starterkit und die Anmeldeinformationen für das Chef Automate-Dashboard herunter. Da Sie dies nicht tun können, wenn Sie einen AWS OpsWorks for Chef Automate Server mithilfe von erstellen AWS CLI, verwenden Sie ein JSON-Verarbeitungsprogramm, um die Anmeldeinformationen und das Starterkit aus den Ergebnissen des `create-server` Befehls abzurufen, nachdem Ihr neuer AWS OpsWorks for Chef Automate Server online ist. Alternativ können Sie neue Anmeldeinformationen und ein neues Starter Kit in der Konsole generieren, nachdem der neue AWS OpsWorks for Chef Automate -Server online ist.

Wenn auf Ihrem lokalen Computer das noch nicht ausgeführt wird AWS CLI, laden Sie es herunter und installieren Sie es, AWS CLI indem Sie den Installationsanweisungen im AWS-Benutzerhandbuch für die Befehlszeilenschnittstelle folgen. In diesem Abschnitt werden nicht alle Parameter beschrieben, die Sie mit dem Befehl `create-server` verwenden können. Weitere Informationen zu den `create-server`-Parametern finden Sie unter `create-server` in der AWS CLI -Referenz.

1. Stellen Sie sicher, dass Sie die Voraussetzungen, insbesondere Einrichten einer VPC erfüllen, oder dass Sie über eine VPC verfügen, die Sie verwenden möchten. Zum Erstellen Ihres Chef Automate-Servers benötigen Sie eine Subnetz-ID.
2. Generieren Sie optional einen pivotalen Chef-Schlüssel mit OpenSSL und speichern Sie den Schlüssel in einer sicheren Datei auf Ihrem lokalen Computer. Der pivotale Schlüssel wird im

Rahmen der Servererstellung automatisch generiert, wenn Sie keinen im Befehl `create-server` angeben. Wenn Sie diesen Schritt überspringen möchten, können Sie stattdessen einen pivotalen Chef Automate-Schlüssel aus den Ergebnissen des Befehls `create-server` extrahieren. Wenn Sie den Zentralschlüssel mithilfe der folgenden Befehle generieren, schließen Sie unbedingt den Parameter `-pubout` ein, da der Wert des Chef Automate-Zentralschlüssels die öffentliche Hälfte des RSA-Schlüsselpaares ist. Weitere Informationen finden Sie in Schritt 6.

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

3. Erstellen Sie eine Servicerolle und ein Instanzprofil. AWS OpsWorks stellt eine AWS CloudFormation Vorlage bereit, mit der Sie beide erstellen können. Führen Sie den folgenden AWS CLI Befehl aus, um einen AWS CloudFormation Stack zu erstellen, der die Servicerolle und das Instanzprofil für Sie erstellt.

```
aws cloudformation create-stack --stack-name OpsWorksCMRoles --template-url
 https://s3.amazonaws.com/opsworks-cm-us-east-1-prod-default-assets/misc/opsworks-
cm-roles.yaml --capabilities CAPABILITY_NAMED_IAM
```

4. Suchen Sie nach AWS CloudFormation Abschluss der Erstellung des Stacks die ARNs der Servicerollen in Ihrem Konto und kopieren Sie sie.

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

Suchen Sie in den Ergebnissen des Befehls `list-roles` nach Servicerollen-ARN-Einträgen, die dem folgenden ähneln. Notieren Sie sich die Servicerollen-ARNs. Sie benötigen diese Werte zum Erstellen Ihres Chef Automate-Servers.

```
{
    "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": "AROZZZZZZZZZZZZZZ6QE",
    "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. Suchen und kopieren Sie die ARNs von Instance-Profilen in Ihrem Konto.

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

Suchen Sie in den Ergebnissen des Befehls `list-instance-profiles` nach Instance-Profil-ARN-Einträgen, die dem folgenden ähneln. Notieren Sie sich die ARNs der Instance-Profile. Sie benötigen diese Werte zum Erstellen Ihres Chef Automate-Servers.

```
{
    "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. Erstellen Sie den AWS OpsWorks for Chef Automate Server, indem Sie den `create-server` Befehl ausführen.

   - Der Wert `--engine` ist `ChefAutomate`, `--engine-model` ist `Single` und `--engine-version` ist 12.
   - Der Servername muss innerhalb Ihres AWS Kontos in jeder Region eindeutig sein. Servernamen müssen mit einem Buchstaben beginnen. Danach können Buchstaben, Zahlen und Bindestriche (-) verwendet werden, insgesamt höchstens 40 Zeichen.
   - Verwenden Sie die ARNs des Instance-Profils und der Servicerolle, die Sie in Schritt 4 und 5 kopiert haben.
   - Gültige Instance-Typen sind `m5.large`, `r5.xlarge` und `r5.2xlarge`. Weitere Informationen zu den Spezifikationen dieser Instance-Typen finden Sie unter Instance-Typen im Amazon EC2 EC2-Benutzerhandbuch.
   - Der Parameter `--engine-attributes` ist optional. Wenn Sie nicht einen oder beide Werte festlegen, werden die Werte bei der Servererstellung für Sie generiert. Wenn Sie `--engine-`

`attributes` hinzufügen, geben Sie entweder den Wert `CHEF_AUTOMATE_PIVOTAL_KEY`, den Sie in Schritt 2 generiert haben, ein `CHEF_AUTOMATE_ADMIN_PASSWORD` oder beides an.

Wenn Sie keinen Wert für `CHEF_AUTOMATE_ADMIN_PASSWORD` festlegen, wird ein Passwort für Sie generiert und in der Antwort des Befehls `create-server` zurückgegeben. Sie können auch das Starter Kit in der Konsole erneut herunterladen, um dieses Passwort erneut zu generieren. Das Passwort muss eine Länge zwischen 8 und 32 Zeichen haben. Das Passwort kann Buchstaben, Zahlen und Sonderzeichen (`!/@#$%^+=_`) enthalten. Es muss mindestens einen Kleinbuchstaben, einen Großbuchstaben, eine Zahl und ein Sonderzeichen enthalten.

- Ein SSH-Schlüsselpaar ist optional. Es kann Ihnen dabei helfen, sich mit dem Chef Automate-Server zu verbinden, wenn Sie das Administratorpasswort des Chef Automate-Dashboards zurücksetzen müssen. Weitere Informationen zum Erstellen eines SSH-Schlüsselpaars finden Sie unter Amazon EC2 EC2-Schlüsselpaare im Amazon EC2 EC2-Benutzerhandbuch.
- Um eine benutzerdefinierte Domäne zu verwenden, fügen Sie dem Befehl die folgenden Parameter hinzu. Andernfalls generiert der Erstellungsprozess des Chef Automate-Servers automatisch einen Endpunkt für Sie. Alle drei Parameter sind erforderlich, um eine benutzerdefinierte Domäne zu konfigurieren. Informationen zu zusätzlichen Anforderungen für die Verwendung dieser Parameter finden Sie CreateServerin der AWS OpsWorks CM-API-Referenz.
  - `--custom-domain` – Ein optionaler öffentlicher Endpunkt eines Servers, z. B. `https://aws.my-company.com`.
  - `--custom-certificate` – Ein PEM-formatiertes HTTPS-Zertifikat. Der Wert kann ein einzelnes, selbstsigniertes Zertifikat oder eine Zertifikatkette sein.
  - `--custom-private-key` – Ein privater Schlüssel im PEM-Format für die Verbindung mit dem Server mithilfe von HTTPS. Der private Schlüssel darf nicht verschlüsselt sein; er kann nicht durch ein Passwort oder eine Passphrase geschützt werden.
- Es ist eine wöchentliche Systemwartung erforderlich. Gültige Werte müssen im folgenden Format angegeben werden: `DDD:HH:MM`. Die angegebene Uhrzeit entspricht der Zeitzone UTC (Coordinated Universal Time). Wenn Sie für `--preferred-maintenance-window` keinen Wert angeben, wird ein zufälliger Standardwert mit einem einstündigen Zeitraum an einem Dienstag, Mittwoch oder Freitag festgelegt.
- Gültige Werte für `--preferred-backup-window` müssen in einem der folgenden Formate angegeben werden: `HH:MM` für tägliche Sicherungen oder `DDD:HH:MM` für wöchentliche Sicherungen. Die angegebene Uhrzeit entspricht der Zeitzone UTC. Standardmäßig wird ein

zufälliger täglicher Startzeitpunkt festgelegt. Wenn Sie automatische Sicherungen deaktivieren möchten, verwenden Sie stattdessen den Parameter `--disable-automated-backup`.

- Geben Sie für `--security-group-ids` eine oder mehrere Sicherheitsgruppen-IDs, durch Kommata getrennt, ein.
- Geben Sie für `--subnet-ids` eine Subnetz-ID ein.

```
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
```

Im Folgenden wird ein Beispiel gezeigt.

```
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
```

Im folgenden Beispiel wird ein Chef Automate-Server erstellt, der eine benutzerdefinierte Domäne verwendet.

```
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
```

Im folgenden Beispiel wird ein Chef Automate-Server erstellt, der zwei Tags hinzufügt: `Stage: Production` und `Department: Marketing`. Weitere Informationen zum Hinzufügen und Verwalten von Tags auf AWS OpsWorks for Chef Automate Servern finden Sie Mit Tags auf AWS OpsWorks for Chef Automate Ressourcen arbeiten in diesem Handbuch.

```
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 dauert etwa 15 Minuten, um einen neuen Server zu erstellen. Sie sollten die Ausgabe des Befehls `create-server` nicht verwerfen oder die Shell-Sitzung beenden, da die Ausgabe wichtige Informationen enthalten kann, die nicht wiederhergestellt werden können. Um Passwörter und das Starter Kit aus den Ergebnissen von `create-server` zu extrahieren, fahren Sie mit dem nächsten Schritt fort.

Wenn Sie eine benutzerdefinierte Domäne mit dem Server verwenden, kopieren Sie in der Ausgabe des Befehls `create-server` den Wert des Attributs `Endpoint`. Im Folgenden wird ein Beispiel gezeigt.

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

8. Wenn Sie sich dafür entschieden haben, einen Schlüssel und ein Passwort für Sie AWS OpsWorks for Chef Automate generieren zu lassen, können Sie diese mithilfe eines JSON-Prozessors wie jq in verwendbaren Formaten aus den `create-server` Ergebnissen extrahieren. Nachdem Sie jq installiert haben, können Sie die folgenden Befehle ausführen, um den pivotalen Schlüssel, das Administratorpasswort für das Chef Automate-Dashboard und das Starter Kit zu extrahieren. Wenn Sie in Schritt 4 keinen eigenen pivotalen Schlüssel und ein eigenes Passwort angegeben haben, speichern Sie den extrahierten pivotalen Schlüssel und das extrahierte Administratorpasswort an einem sicheren Speicherort.

```
#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
```

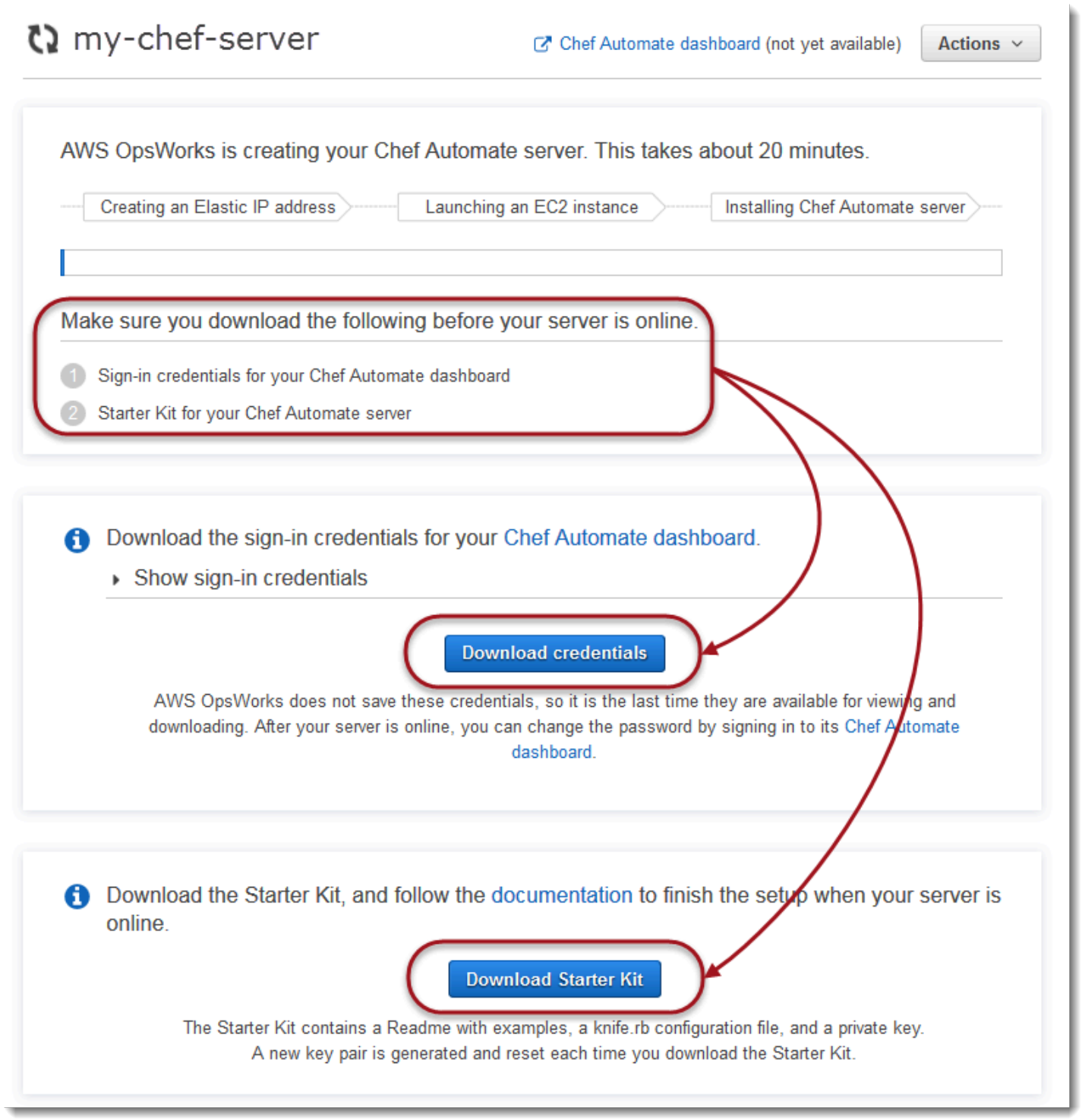
9. Falls Sie das Starterkit nicht aus den `create-server` Befehlsergebnissen extrahiert haben, können Sie optional ein neues Starterkit von der Eigenschaftenseite des Servers in der AWS OpsWorks for Chef Automate Konsole herunterladen. Wenn Sie ein neues Starter Kit herunterladen, wird das Administratorpasswort des Chef Automate-Dashboards zurückgesetzt.
10. Wenn Sie keine benutzerdefinierte Domäne verwenden, fahren Sie mit dem nächsten Schritt fort. Wenn Sie eine benutzerdefinierte Domäne mit dem Server verwenden, erstellen Sie einen CNAME-Eintrag im DNS-Verwaltungstool Ihres Unternehmens, um Ihre benutzerdefinierte Domäne auf den AWS OpsWorks for Chef Automate Endpunkt zu verweisen, den Sie in Schritt 7 kopiert haben. Sie können einen Server erst dann mit einer benutzerdefinierten Domäne erreichen und sich erst dann bei ihm anmelden, nachdem Sie diesen Schritt ausgeführt haben.

11. Fahren Sie nach Abschluss der Servererstellung mit the section called “Konfiguration abschließen und Rezeptbuch hochladen” fort.

Konfigurieren des Chef-Servers mit dem Starter Kit

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

Öffnen Sie während der Erstellung des Chef-Servers in der AWS OpsWorks for Chef Automate - Konsole die Seite Eigenschaften. Wenn Sie zum ersten Mal mit einem neuen Chef-Server arbeiten, werden Sie von der Eigenschaftenseite aufgefordert, die erforderlichen Elemente herunterzuladen. Laden Sie diese Elemente herunter, bevor Ihr Chef-Server online ist. Die Schaltflächen zum Herunterladen sind nicht verfügbar, wenn ein neuer Server online ist.



- Anmeldeinformationen für den Chef-Server. Sie verwenden diese Anmeldeinformationen, um sich beim Chef Automate-Dashboard anzumelden, wo Sie mit den Premium-Funktionen von Chef Automate wie Workflow- und Compliance-Scans arbeiten. AWS OpsWorks speichert diese Anmeldeinformationen nicht; dies ist das letzte Mal, dass sie zum Anzeigen und Herunterladen

verfügbar sind. Falls erforderlich, können Sie das Passwort ändern, nachdem Sie sich mit diesen Anmeldeinformationen angemeldet haben.

- Starter Kit. Das Starter Kit enthält eine Readme-Datei mit Beispielen, die Konfigurationsdatei `knife.rb` und einen privaten Schlüssel für den primären oder pivotalen Benutzer. Jedes Mal, wenn Sie das Starter Kit herunterladen, wird ein neues Schlüsselpaar erstellt und der alte Schlüssel zurückgesetzt.

Zusätzlich zu den Anmeldeinformationen, die nur mit dem neuen Server funktionieren, enthält die Starter Kit-ZIP-Datei ein einfaches Beispiel für ein Chef-Repository, das mit jedem AWS OpsWorks for Chef Automate Server funktioniert. Im Chef-Repository werden Rezeptbücher, Rollen, Konfigurationsdateien und andere Artefakten zur Verwaltung Ihrer Knoten mit Chef gespeichert. Wir empfehlen, dass Sie dieses Repository in einem Versionskontrollsystem wie z. B. Git speichern und als Quellcode behandeln. Weitere Informationen und Beispiele, die verdeutlichen, wie ein in Git nachverfolgtes Chef-Repository erstellt wird, finden Sie in der Chef-Dokumentation unter About the chef-repo.

Voraussetzungen

1. Laden Sie während der Servererstellung die Anmeldeinformationen für den Chef-Server herunter und speichern Sie diese an einem sicheren, aber leicht zugreifbaren Ort.
2. Laden Sie das Starter Kit herunter und extrahieren Sie die .zip-Datei des Starter Kits in Ihr Workspace-Verzeichnis. Teilen Sie nicht den privaten Schlüssel des Starter Kits. Wenn andere Benutzer den Chef-Server verwalten, fügen Sie diese zu einem späteren Zeitpunkt als Administratoren im Chef Automate-Dashboard hinzu.
3. Laden Sie Chef Workstation (früher bekannt als Chef Development Kit oder Chef DK) herunter und installieren Sie es auf dem Computer, den Sie zur Verwaltung Ihres Chef-Servers und Ihrer Chef-Knoten verwenden werden. Das `knife`Hilfsprogramm ist Teil von Chef Workstation. Anweisungen finden Sie unter Chef Workstation installieren auf der Chef-Website.

Erkunden des Inhalts vom Starter Kit

Das Starter Kit umfasst die folgenden Inhalte.

- `cookbooks/` – Ein Verzeichnis für Rezeptbücher, die Sie erstellen. Der `cookbooks/` Ordner enthält das `opsworks-webserver` Kochbuch, ein Wrapper-Kochbuch, das vom Kochbuch auf der Chef `nginx` Supermarkt-Website abhängt. `Policyfile.rb`verwendet standardmäßig Chef

Supermarket als sekundäre Quelle, wenn Kochbuchabhängigkeiten im Verzeichnis nicht verfügbar sind. `cookbooks/`

- `Policyfile.rb` – Eine auf Ruby basierte Richtliniendatei, mit der die Rezeptbücher, Abhängigkeiten und Attribute definiert werden, die als Richtlinie für Ihre Knoten fungieren.
- `userdata.sh` und `userdata.ps1` – Sie können mit Benutzerdatendateien Knoten automatisch verknüpften, nachdem Sie Ihrem Chef Automate-Server gestartet haben. `userdata.sh` dient dem Bootstrapping Linux-basierter Knoten und `userdata.ps1` wird für Windows-basierte Knoten verwendet.
- `Berksfile` – Sie können diese Datei verwenden, wenn Sie Berkshelf verwenden möchten und Rezeptbücher sowie deren Abhängigkeiten mit den `berks`-Befehlen hochladen. Im Rahmen dieser Anleitung verwenden wir `Policyfile.rb` und Chef-Befehle zum Hochladen von Rezeptbüchern, Abhängigkeiten und Attributen.
- `README.md`, eine Markdown-basierte Datei, mit der die Verwendung des Starter Kits für die erstmalige Einrichtung Ihrer Chef Automate-Server beschrieben wird.
- `.chef` ist ein ausgeblendetes Verzeichnis mit einer Knife-Konfigurationsdatei (`knife.rb`) und einer geheimen Authentifizierungsschlüsseldatei (.pem).
  - `.chef/knife.rb` – Eine Knife-Konfigurationsdatei (`knife.rb`). Die `knife.rb`Datei ist so konfiguriert, dass die `knife`Tooloperationen von Chef auf dem AWS OpsWorks for Chef Automate Server ausgeführt werden.
  - `.chef/ca_certs/opsworks-cm-ca-2020-root.pem` – Ein von einer Zertifizierungsstelle (Certification Authority, CA) signierter privater SSL-Schlüssel, der von AWS OpsWorks bereitgestellt wird. Mit diesem Schlüssel kann sich der Server gegenüber dem Chef-Infra-Agenten auf den von Ihrem Server verwalteten Knoten identifizieren.

Einrichten des Chef-Repositorys

Ein Chef-Repository enthält mehrere Verzeichnisse. Jedes Verzeichnis im Starter Kit enthält eine Readme-Datei, die den Zweck des Verzeichnisses beschreibt und wie es für die Verwaltung Ihrer Systeme mit Chef verwendet wird. Es gibt zwei Möglichkeiten für die Installation von Rezeptbüchern auf Ihrem Chef-Server: mit `knife`-Befehlen oder über einen Chef-Befehl zum Hochladen einer Richtliniendatei (`Policyfile.rb`) auf Ihren Server, der bestimmte Rezeptbücher herunterlädt und installiert. In dieser Anleitung werden Chef-Befehle und `Policyfile.rb` zum Installieren von Rezeptbüchern auf Ihrem Server verwendet.

1. Erstellen Sie auf Ihrem lokalen Computer ein Verzeichnis zum Speichern von Rezeptbüchern, z. B. `chef-repo`. Nachdem Sie Kochbücher, Rollen und andere Dateien zu diesem Repository hinzugefügt haben, empfehlen wir, diese hochzuladen oder in einem sicheren, versionierten System wie CodeCommit Git oder Amazon S3 zu speichern.
2. Erstellen Sie im `chef-repo`-Verzeichnis die folgenden Verzeichnisse:
   - `cookbooks/`- Speichert Kochbücher.
   - `roles/` – Speichert Rollen im `.rb`- oder `.json`-Format.
   - `environments/` – Speichert Umgebungen im `.rb`- oder `.json`-Format.

Verwenden von Policyfile.rb zum Abrufen von Rezeptbüchern aus einer externen Quelle

Bearbeiten Sie in diesem Abschnitt `Policyfile.rb`, um Rezeptbücher anzugeben, laden Sie die Datei mit einem Chef-Befehl auf Ihrem Server hoch und installieren Sie Rezeptbücher.

1. Zeigen Sie `Policyfile.rb` in Ihrem Starter Kit an. Standardmäßig enthält `Policyfile.rb` das `opsworks-webserver`-Wrapper-Rezeptbuch. Das hängt vom `nginx`-Rezeptbuch ab, das Sie auf Chef Supermarket-Website finden. Das `nginx`-Rezeptbuch installiert und konfiguriert einen Webserver auf verwalteten Knoten. Das erforderliche `chef-client`-Rezeptbuch, das den Chef Infra-Client-Agenten auf verwalteten Knoten installiert, wird ebenfalls angegeben.

   Darüber hinaus weist `Policyfile.rb` auf das optionale Chef Audit-Rezeptbuch, mit dem Sie Compliance-Scans auf Knoten einrichten können. Weitere Informationen zum Einrichten von Compliance-Scans und zum Abrufen von Compliance-Ergebnissen für verwaltete Knoten finden Sie unter Konformitätsscans in AWS OpsWorks for Chef Automate. Wenn Sie Compliance-Scans und Auditing nicht sofort konfigurieren möchten, löschen Sie `'audit'` aus dem `run_list`-Abschnitt und geben Sie keine `audit`-Rezeptbuchattribute am Ende der Datei an.

```
# 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"
  }
]
```

Es folgt ein Beispiel für `Policyfile.rb` ohne das `audit`-Rezeptbuch und Attribute, wenn Sie erst einmal nur den `nginx`-Webserver konfigurieren möchten.

```
# 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'
```

Wenn Sie Änderungen an `Policyfile.rb` vornehmen, stellen Sie sicher, die Datei zu speichern.

2. Laden Sie die in `Policyfile.rb` definierten Rezeptbücher herunter und installieren Sie sie.

```
chef install
```

Alle Rezeptbücher sind in der `metadata.rb`-Datei des Rezeptbuchs versioniert. Immer wenn Sie ein Rezeptbuch ändern, müssen Sie die Version des Rezeptbuchs ändern, die sich in seiner `metadata.rb` befindet.

3. Wenn Sie Compliance-Scans konfigurieren möchten und die `audit`-Rezeptbuchinformationen in die Richtliniendatei belassen haben, leiten Sie die `opsworks-demo`-Richtlinie an Ihren Server weiter.

```
chef push opsworks-demo
```

4. Wenn Sie Schritt 3 abgeschlossen haben, überprüfen Sie die Installation Ihrer Richtlinie. Führen Sie den folgenden Befehl aus.

```
chef show-policy
```

Die Ergebnisse sollten etwa wie folgt aussehen:

```
opsworks-demo-webserver
```

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

5. Sie können Ihrem Chef Automate-Server nun Knoten hinzufügen oder Bootstrapping dafür durchführen. Sie können die Zuordnung von Knoten automatisieren, indem Sie die Schritte in Automatisches Hinzufügen von Knoten AWS OpsWorks for Chef Automate ausführen, oder Knoten einzeln hinzufügen, indem Sie die Schritte in Fügen Sie Knoten einzeln hinzu ausführen.

(Alternativ) Verwenden von Berkshelf zum Abrufen von Rezeptbüchern aus einer externen Quelle

Berkshelf ist ein Tool für die Verwaltung von Rezeptbüchern und deren Abhängigkeiten. Wenn Sie die Verwendung von Berkshelf anstelle von `Policyfile.rb` vorziehen, um Rezeptbücher im lokalen Speicher zu installieren, verwenden Sie das Verfahren in diesem Abschnitt anstelle der Verfahrens im vorherigen Abschnitt. Sie können angeben, welche Rezeptbücher und Versionen Sie mit Ihren Chef-Server verwenden möchten, und sie hochladen. Das Starter Kit enthält eine Datei namens `Berksfile`, mit dem Sie Ihre Rezeptbücher aufführen können.

1. Um zu beginnen, fügen Sie dem enthaltenen Berksfile das `chef-client`-Rezeptbuch hinzu. Das `chef-client`-Rezeptbuch konfiguriert die -Agent-Software des Chef Infra-Clients auf jedem Knoten, den Sie mit Ihrem Chef Automate-Server verbinden. Weitere Informationen zu diesem Rezeptbuch finden Sie unter Chef Client Cookbook im Chef Supermarket.
2. Hängen Sie Ihrem Berksfile mit einem Texteditor ein anderes Rezeptbuch an, mit dem eine Web-Server-Anwendung installiert wird, z. B. das `apache2`-Rezeptbuch, das den Apache-Webserver installiert. Ihr Berksfile sollte etwa folgendermaßen aussehen.

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

3. Laden Sie die Rezeptbücher herunter und installieren Sie sie auf Ihrem lokalen Computer.

   ```
   berks install
   ```

4. Laden Sie das Rezeptbuch auf den Chef-Server hoch.

   Führen Sie unter Linux Folgendes aus.

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

Führen Sie unter Windows den folgenden Chef Workstation-Befehl in einer PowerShell Sitzung aus. Bevor Sie den Befehl ausführen, stellen Sie sicher, dass die Ausführungsrichtlinie PowerShell auf eingestellt ist`RemoteSigned`. Fügen Sie hinzu`chef shell-init`, um die Befehle des Chef Workstation-Dienstprogramms für verfügbar zu machen 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. Überprüfen Sie die Installation des Rezeptbuchs, indem Sie eine Liste der Rezeptbücher anzeigen, die gegenwärtig auf dem Chef Automate-Server verfügbar sind. Sie erreichen dies mit dem `knife`-Befehl-

   Sie sind bereit, Knoten hinzuzufügen, die mit dem AWS OpsWorks for Chef Automate Server verwaltet werden sollen.

   ```
   knife cookbook list
   ```

(Optional) Konfigurieren von **`knife`** zur Verwendung mit einer benutzerdefinierten Domäne

Wenn Ihr Chef Automate-Server eine benutzerdefinierte Domäne verwendet, müssen Sie möglicherweise das PEM-Zertifikat der Stammzertifizierungsstelle, die die Zertifikatkette Ihres Servers signiert hat, oder das PEM-Zertifikat des Servers hinzufügen, wenn das Zertifikat selbstsigniert ist. `ca_certs` ist ein Unterverzeichnis in `chef/`, das Zertifizierungsstellen (CAs) enthält, denen das Chef-Dienstprogramm `knife` vertraut.

Sie können diesen Abschnitt überspringen, wenn Sie keine benutzerdefinierte Domäne verwenden oder wenn Ihr benutzerdefiniertes Zertifikat von einer Stammzertifizierungsstelle signiert ist, der Ihr Betriebssystem vertraut. Andernfalls konfigurieren Sie `knife` wie in den folgenden Schritten beschrieben, sodass dem SSL-Zertifikat Ihres Chef Automate-Servers vertraut wird.

1. Führen Sie den folgenden Befehl aus.

   ```
   knife ssl check
   ```

Wenn die Ergebnisse den folgenden ähneln, überspringen Sie den Rest dieses Verfahrens, und fahren Sie mit Hinzufügen von Knoten, die vom Chef-Server verwaltet werden fort.

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

Wenn Sie eine Fehlermeldung ähnlich der folgenden erhalten, fahren Sie mit dem nächsten Schritt fort.

```
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. Führen Sie `knife ssl fetch` aus, um den Zertifikaten Ihres AWS OpsWorks for Chef Automate -Servers zu vertrauen. Alternativ können Sie das PEM-formatierte Stammzertifikat des Servers manuell in das Verzeichnis kopieren, das dem Wert `trusted_certs_dir` in der Ausgabe von `knife ssl check` entspricht. Standardmäßig befindet sich dieses Verzeichnis in `.chef/ca_certs/` im Starter Kit. Die Ausgabe sollte in etwa wie folgt aussehen:

```
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. Führen Sie `knife ssl check` erneut aus. Die Ausgabe sollte in etwa wie folgt aussehen:

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

Sie können `knife` mit Ihrem Chef Automate-Server verwenden.

Hinzufügen von Knoten, die vom Chef-Server verwaltet werden

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Lebensende erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

Der `chef-client`-Agent führt Chef-Rezepte auf physischen oder virtuellen Computern, sogenannte Knoten, aus, die mit dem Server verknüpft sind. Sie können zum Verwalten lokale Computer oder Instances mit dem Chef-Server verbinden, sofern die Knoten mit einem unterstützten Betriebssystem laufen. Durch das Registrieren von Knoten mit dem Chef-Server wird auf diesen Knoten die `chef-client`-Agent-Software installiert.

Sie können die folgenden Methoden verwenden, um Knoten hinzuzufügen:

- Fügen Sie Notizen einzeln hinzu, indem Sie einen `knife` Befehl ausführen, der eine EC2-Instanz hinzufügt oder bootet, sodass der Chef-Server sie verwalten kann. Weitere Informationen finden Sie unter Fügen Sie Knoten einzeln hinzu.
- Fügen Sie Knoten automatisch hinzu, indem Sie ein Skript verwenden, um die Knoten unbeaufsichtigt mit dem Chef-Server zu verknüpfen. Der Code in diesem Starter Kit zeigt das automatische Hinzufügen von Knoten mit der unbeaufsichtigten Methode. Weitere Informationen finden Sie unter Automatisches Hinzufügen von Knoten AWS OpsWorks for Chef Automate.

Themen

Fügen Sie Knoten einzeln hinzu

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Lebensende erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

In diesem Abschnitt wird beschrieben, wie Sie einen `knife` Befehl ausführen, der eine EC2-Instance hinzufügt oder bootet, sodass der Chef-Server sie verwalten kann.

Die unterstützte Mindestversion von `chef-client` auf mit einem AWS OpsWorks for Chef Automate -Server verknüpften Knoten ist 13x. Wir empfehlen, die aktuellste, `chef-client` stabilste Version auszuführen.

Themen

(Optional) Angeben der URL Ihrer Chef Automate-Server-Root-CA

Wenn Ihr Server eine benutzerdefinierte Domäne und ein eigenes Zertifikat verwendet, müssen Sie die Variable `ROOT_CA_URL` im Benutzerdatenskript möglicherweise mit einer öffentlichen URL bearbeiten, mit der Sie das PEM-formatierte Stammzertifikat des Servers abrufen können. Die folgenden AWS CLI Befehle laden Ihre Root-CA in einen Amazon S3 S3-Bucket hoch und generieren eine vorsignierte URL, die Sie eine Stunde lang verwenden können.

1. Laden Sie das PEM-formatierte Stammzertifikat der Zertifizierungsstelle in S3 hoch.

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

2. Generieren Sie eine vorsignierte URL, die Sie für eine Stunde (in diesem Beispiel 3600 Sekunden) verwenden können, um die Stammzertifizierungsstelle herunterzuladen.

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

3. Bearbeiten Sie die Variable ROOT_CA_URL im Benutzerdatenskript mit dem Wert der vorsignierten URL.

Unterstützte Betriebssysteme

Eine aktuelle Liste von unterstützten Betriebssystemen für Knoten finden Sie auf der Chef-Website.

Hinzufügen von Knoten mit Knife

Das `knife-ec2`Plug-in ist in Chef Workstation enthalten. Wenn Sie mit `knife-ec2` vertraut sind, können Sie es anstelle von `knife bootstrap` für die Bereitstellung und den Bootstrap neuer EC2-Instances verwenden. Starten Sie andernfalls eine neue EC2-Instance und führen Sie dann die in diesem Abschnitt beschriebenen Schritte aus.

So fügen Sie zu verwaltende Knoten hinzu

1. Führen Sie den Befehl `knife bootstrap` aus. Mit diesem Befehl erfolgt das Bootstrapping einer EC2-Instance an die Knoten, die von Ihrem Chef-Server verwaltet werden. Beachten Sie, dass Sie den Chef-Server anweisen, die Rezepte des `nginx`-Rezeptbuchs auszuführen, das Sie in the section called “Verwenden von Policyfile.rb zum Abrufen von Rezeptbüchern aus einer externen Quelle” installiert haben. Weitere Informationen über das Hinzufügen von Knoten, indem Sie den `knife bootstrap`-Befehl ausführen, finden Sie in der Chef-Dokumentation unter Bootstrap a Node.

   Die folgende Tabelle enthält die gültigen Benutzernamen für Knoten-Betriebssysteme im `knife`-Befehl in diesem Schritt. Wenn weder `root` noch `ec2-user` funktioniert, wenden Sie sich an Ihren AMI-Anbieter. Weitere Informationen zum Herstellen einer Verbindung mit Linux-basierten Instances finden Sie in der AWS-Dokumentation unter Herstellen einer Verbindung mit Ihrer Linux-Instance per SSH.

   Gültige Werte für Benutzernamen in Knoten-Betriebssystemen

Betriebssystem	Zulässige Benutzernamen
Amazon Linux	`ec2-user`
Red Hat Enterprise Linux 5	`root` oder `ec2-user`

Betriebssystem	Zulässige Benutzernamen
Ubuntu	`ubuntu`
Fedora	`fedora` oder `ec2-user`
SUSE Linux	`root` oder `ec2-user`

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

2. Überprüfen Sie, ob der neue Knoten hinzugefügt wurde, indem Sie die folgenden Befehle ausführen, wobei *INSTANCE_NAME* durch den Namen der Instance, die Sie gerade hinzugefügt haben, zu ersetzen ist.

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

Automatisches Hinzufügen von Knoten AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

In diesem Thema wird beschrieben, wie Sie Ihrem Chef-Server automatisch Amazon Elastic Compute Cloud (Amazon EC2) -Knoten hinzufügen. Der Code in diesem Starter Kit zeigt das automatische Hinzufügen von Knoten mit der unbeaufsichtigten Methode. Die empfohlene Methode für eine unbeaufsichtigte (oder automatische) Zuordnung neuer Knoten besteht in der Konfiguration des Chef Client Cookbook. Sie können das Skript `userdata` im Starter Kit verwenden und entweder den `run_list`-Abschnitt des Skripts `userdata` oder Ihr `Policyfile.rb` mit den Rezeptbüchern ändern, die Sie auf Ihrem Knoten anwenden möchten. Bevor Sie den `chef-client`-Agenten

ausführen, laden Sie das Chef Client-Rezeptbuch auf Ihren Chef-Server hoch und installieren den `chef-client`-Agenten im Service-Modus, beispielsweise mit einer HTTPD-Rolle, wie im folgenden Beispielbefehl veranschaulicht.

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

Für die Kommunikation mit dem Chef-Server benötigt die `chef-client`-Agentensoftware Zugriff auf den öffentlichen Schlüssel des Client-Knotens. Sie können in Amazon EC2 ein öffentlich-privates key pair generieren und dann den öffentlichen Schlüssel mit dem Knotennamen an den AWS OpsWorks `associate-node` API-Aufruf übergeben. Das Skript in diesem Starter Kit fasst den Namen der Organisation, den Servernamen und den Server-Endpunkt für Sie zusammen. So wird sichergestellt, dass der Knoten dem Chef-Server zugeordnet wird und die `chef-client`-Agentensoftware, die auf dem Knoten ausgeführt wird, bei einer Übereinstimmung mit dem privaten Schlüssel mit dem Server kommunizieren kann.

Die unterstützte Mindestversion von `chef-client` auf mit einem AWS OpsWorks for Chef Automate -Server verknüpften Knoten ist 13x. Wir empfehlen, die aktuellste, stabilste Version zu verwenden. `chef-client`

Informationen zum Trennen der Zuordnung eines Knotens finden Sie Einen Knoten von einem AWS OpsWorks for Chef Automate Server trennen in diesem Handbuch und `disassociate-node`in der AWS OpsWorks for Chef Automate API-Dokumentation.

Themen

Unterstützte Betriebssysteme

Eine aktuelle Liste von unterstützten Betriebssystemen für Knoten finden Sie auf der Chef-Website.

Schritt 1: Erstellen Sie eine IAM-Rolle, die Sie als Ihr Instanzprofil verwenden möchten

Erstellen Sie eine AWS Identity and Access Management (IAM-) Rolle, die Sie als Ihr EC2-Instance-Profil verwenden möchten, und fügen Sie der IAM-Rolle die folgende Richtlinie hinzu. Diese Richtlinie ermöglicht der AWS OpsWorks for Chef Automate -API (`opsworks-cm`) die Kommunikation mit der EC2-Instance während der Knotenregistrierung. Weitere Informationen zu Instance-Profilen finden Sie unter Using Instance Profiles in der Amazon EC2 EC2-Dokumentation. Informationen zum Erstellen einer IAM-Rolle finden Sie unter Creating an IAM Role in the Console in der Amazon EC2 EC2-Dokumentation.

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

AWS OpsWorks stellt eine AWS CloudFormation Vorlage bereit, die Sie verwenden können, um die IAM-Rolle mit der vorherigen Richtlinienerklärung zu erstellen. Der folgende AWS CLI Befehl erstellt mithilfe dieser Vorlage die Instanzprofilrolle für Sie. Sie können den `--region` Parameter weglassen, wenn Sie den neuen AWS CloudFormation Stack in Ihrer Standardregion erstellen möchten.

```
aws cloudformation --region region ID create-stack --stack-
name myChefAutomateinstanceprofile --template-url https://s3.amazonaws.com/opsworks-
cm-us-east-1-prod-default-assets/misc/opsworks-cm-nodes-roles.yaml --capabilities
 CAPABILITY_IAM
```

Schritt 2: Installieren des Chef Client-Rezeptbuchs

Sofern dies noch nicht geschehen ist, führen Sie die Schritte in (Alternativ) Verwenden von Berkshelf zum Abrufen von Rezeptbüchern aus einer externen Quelle aus, um sicherzustellen, dass Ihr

Berksfile oder Ihre `Policyfile.rb`-Datei auf das Chef Client-Rezeptbuch verweist und dieses installiert.

Schritt 3: Erstellen von Instances mit einem unbeaufsichtigten Skript für die Zuordnung

1. Um EC2-Instances zu erstellen, können Sie das `userdata` Skript aus dem Starter Kit in den `userdata` Abschnitt mit EC2-Instance-Anweisungen, Amazon EC2 Auto Scaling Scaling-Gruppenstartkonfigurationen oder in eine Vorlage kopieren. AWS CloudFormation Weitere Informationen zum Hinzufügen von Skripten zu Benutzerdaten finden Sie unter Running Commands on Your Linux Instance at Launch in der Amazon EC2 EC2-Dokumentation.

   Dieses Skript führt den `opsworks-cm`-API-Befehl `associate-node` aus, um dem Chef-Server einen neuen Knoten zuzuordnen.

   Standardmäßig ist der Name des neu registrierten Knotens die Instance-ID. Sie können den Namen aber ändern, indem Sie den Wert der `NODE_NAME`-Variable im Skript `userdata` ändern. Da das Ändern des Organisationsnamens in der Benutzeroberfläche der Chef-Konsole derzeit nicht möglich ist, belassen Sie die Festlegung von `CHEF_AUTOMATE_ORGANIZATION` auf `default`.
2. Befolgen Sie die Anleitung in Starten einer Instance in der EC2-Dokumentation mit Änderungen. Wählen Sie im EC2-Instance-Startassistenten ein Amazon Linux AMI.
3. Wählen Sie auf der Seite Configure Instance Details (Instance-Details konfigurieren) die Rolle, die Sie in Schritt 1: Erstellen Sie eine IAM-Rolle, die Sie als Ihr Instanzprofil verwenden möchten erstellt haben, als Ihre IAM-Rolle.
4. Laden Sie in den Bereich Advanced Details (Erweiterte Details) das `userdata.sh`-Skript hoch, das Sie zuvor in diesem Verfahren erstellt haben.
5. Auf der Seite Add Storage (Speicher hinzufügen) sind keine Änderungen erforderlich. Gehen Sie weiter zu Add Tags (Tags hinzufügen).
6. Wählen Sie auf der Seite Configure Security Group (Sicherheitsgruppe konfigurieren) Add Rule (Regel hinzufügen) und dann den Typ HTTP, um in diesem Beispiel die Ports 443 und 80 für den Apache-Webserver zu öffnen.
7. Wählen Sie Review and Launch (Überprüfen und starten) und dann Launch (Starten) aus. Wenn Ihr neuer Knoten startet, werden die Konfigurationen angewendet, die von den im `RUN_LIST`-Parameter angegebenen Rezepten vorgegeben sind.

8. Optional: Wenn Sie Ihrer Ausführungsliste das `nginx`-Rezeptbuch hinzugefügt haben und die mit dem öffentlichen DNS Ihres neuen Knotens verknüpfte Webseite öffnen, sollten Sie eine Website sehen, die von Ihrem nginx-Webserver gehostet wird.

Andere Methoden zur Automatisierung wiederholter Ausführungen von **`chef-client`**

Es ist zwar schwieriger zu erreichen und wird auch nicht empfohlen, aber Sie können das Skript in diesem Thema ausschließlich als Teil der Benutzerdaten einer eigenständigen Instanz ausführen, eine AWS CloudFormation Vorlage verwenden, um es zu neuen Instance-Benutzerdaten hinzuzufügen, einen `cron` Job so konfigurieren, dass das Skript regelmäßig ausgeführt wird, oder `chef-client` innerhalb eines Service ausgeführt werden. Wir empfehlen allerdings die Methode des Chef Client-Rezeptbuchs, da einige Nachteile mit anderen Automatisierungstechniken bestehen

Eine vollständige Liste der Parameter, die für `chef-client` bereitgestellt werden können, finden Sie in der Dokumentation zu Chef.

Verwandte Themen

Die folgenden AWS Blogbeiträge bieten weitere Informationen zur automatischen Zuordnung von Knoten zu Ihrem Chef Automate-Server, mithilfe von Auto Scaling Scaling-Gruppen oder innerhalb mehrerer Konten.

- Verwendung von AWS OpsWorks for Chef Automate zur Verwaltung von EC2-Instances mit Auto Scaling
- OpsWorks für Chef Automate — Automatisches Bootstrapping von Knoten in verschiedenen Konten

Anmelden beim Chef Automate-Dashboard

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

Nachdem Sie die Anmeldedaten von der Eigenschaftenseite des Chef-Servers heruntergeladen haben und wenn der Server online ist, melden Sie sich beim Chef Automate-Dashboard an. In dieser Anleitung haben wir Sie gebeten, zunächst Rezeptbücher hochzuladen und mindestens einen zu verwaltenden Knoten hinzuzufügen. Auf diese Weise werden Informationen über die Rezeptbücher und Knoten im Dashboard angezeigt.

Wenn Sie versuchen, eine Verbindung zur Dashboard-Webseite herzustellen, werden Zertifikatswarnungen in Ihrem Browser angezeigt, bis Sie ein AWS OpsWorks spezifisches, von einer Zertifizierungsstelle signiertes SSL-Zertifikat auf dem Client-Computer installieren, den Sie zur Verwaltung Ihres Chef-Servers verwenden. Wenn die Warnungen beim Aufrufen der Dashboard-Webseite nicht erscheinen sollen, installieren Sie das SSL-Zertifikat, bevor Sie sich anmelden.

Um das SSL-Zertifikat zu installieren AWS OpsWorks

- Wählen Sie das für Ihr System geeignete Zertifikat.
  - Laden Sie für Linux- oder macOS-basierte Systeme die Datei mit der PEM-Dateinamenerweiterung vom folgenden Amazon S3 S3-Speicherort herunter: https:// s3.amazonaws.com/ opsworks-cm-us-east -1- prod-default-assets /misc/ -2016-root.pem. opsworks-cm-ca

    > **Note**
    >
    > Laden Sie zusätzlich eine neuere PEM-Datei vom folgenden Speicherort herunter:. https://s3.amazonaws.com/opsworks-cm-us-east-1-prod-default-assets/misc/opsworks-cm-ca-2020-root.pem Da AWS OpsWorks for Chef Automate derzeit die Stammzertifikate erneuert werden, müssen Sie sowohl alten als auch neuen Zertifikaten vertrauen.

    Weitere Informationen zur Verwaltung von SSL-Zertifikaten auf macOS findest du unter Informationen zu einem Zertifikat in Keychain Access auf dem Mac auf der Apple Support-Website abrufen.
  - Für Windows-Systeme laden Sie die Datei mit der Dateinamenerweiterung P7B vom folgenden Amazon S3 S3-Speicherort herunter: https://s3.amazonaws.com/ opsworks-cm-us-east -1 - / misc/ -2016-root.p7b. prod-default-assets opsworks-cm-ca

> **Note**
>
> Laden Sie zusätzlich eine neuere P7B-Datei https://s3.amazonaws.com/opsworks-cm-us-east-1-prod-default-assets/misc/opsworks-cm-ca-2020-root.p7b vom folgenden Speicherort herunter:. Da AWS OpsWorks for Chef Automate derzeit die Stammzertifikate erneuert werden, müssen Sie sowohl alten als auch neuen Zertifikaten vertrauen.

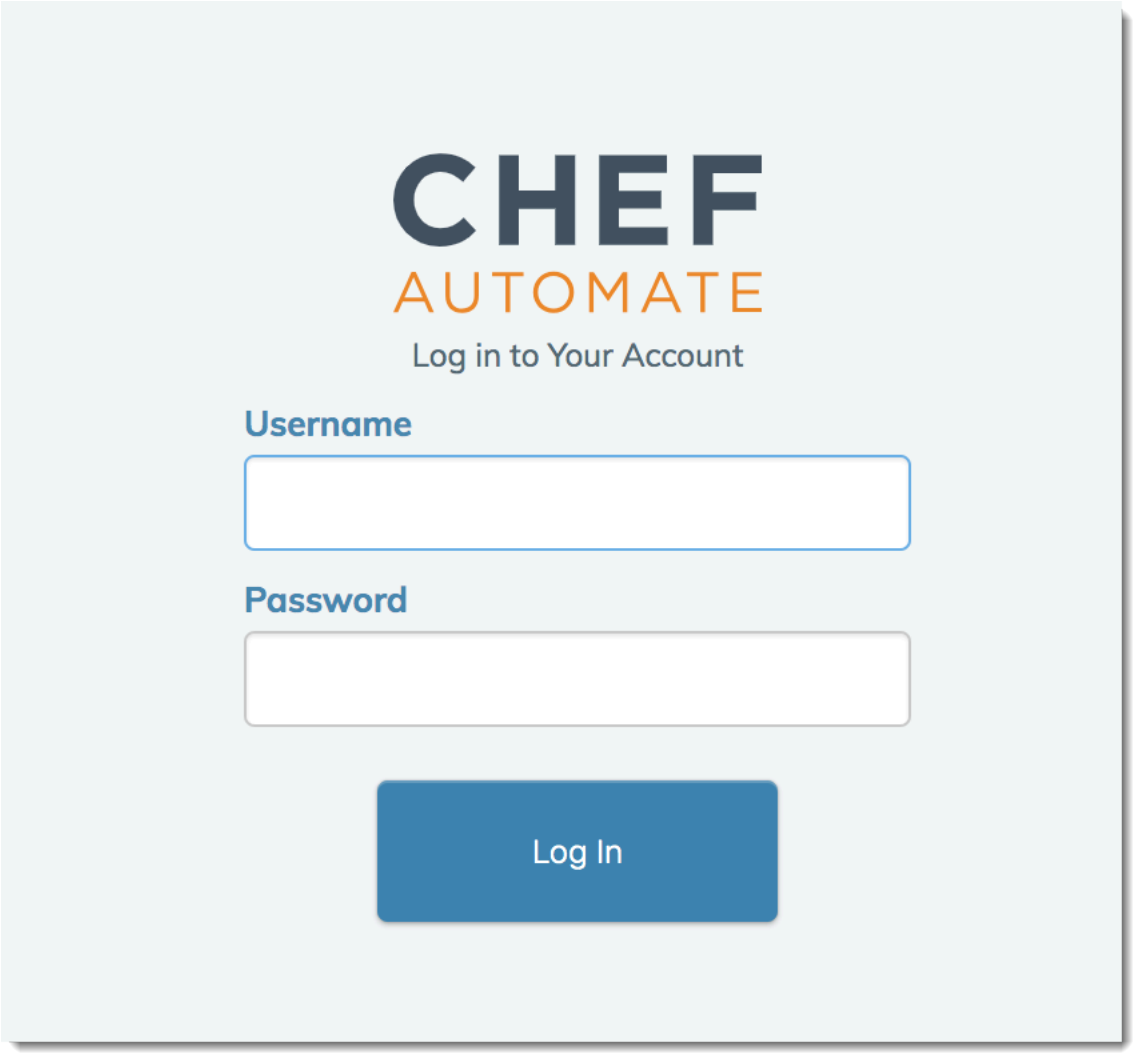
Weitere Informationen zur Installation eines SSL-Zertifikats unter Windows finden Sie unter Vertrauenswürdige Stammzertifikate verwalten auf Microsoft TechNet.

Nachdem Sie das Client-seitige SSL-Zertifikat installiert haben, können Sie sich beim Chef Automate-Dashboard anmelden, ohne dass Warnmeldungen erscheinen.

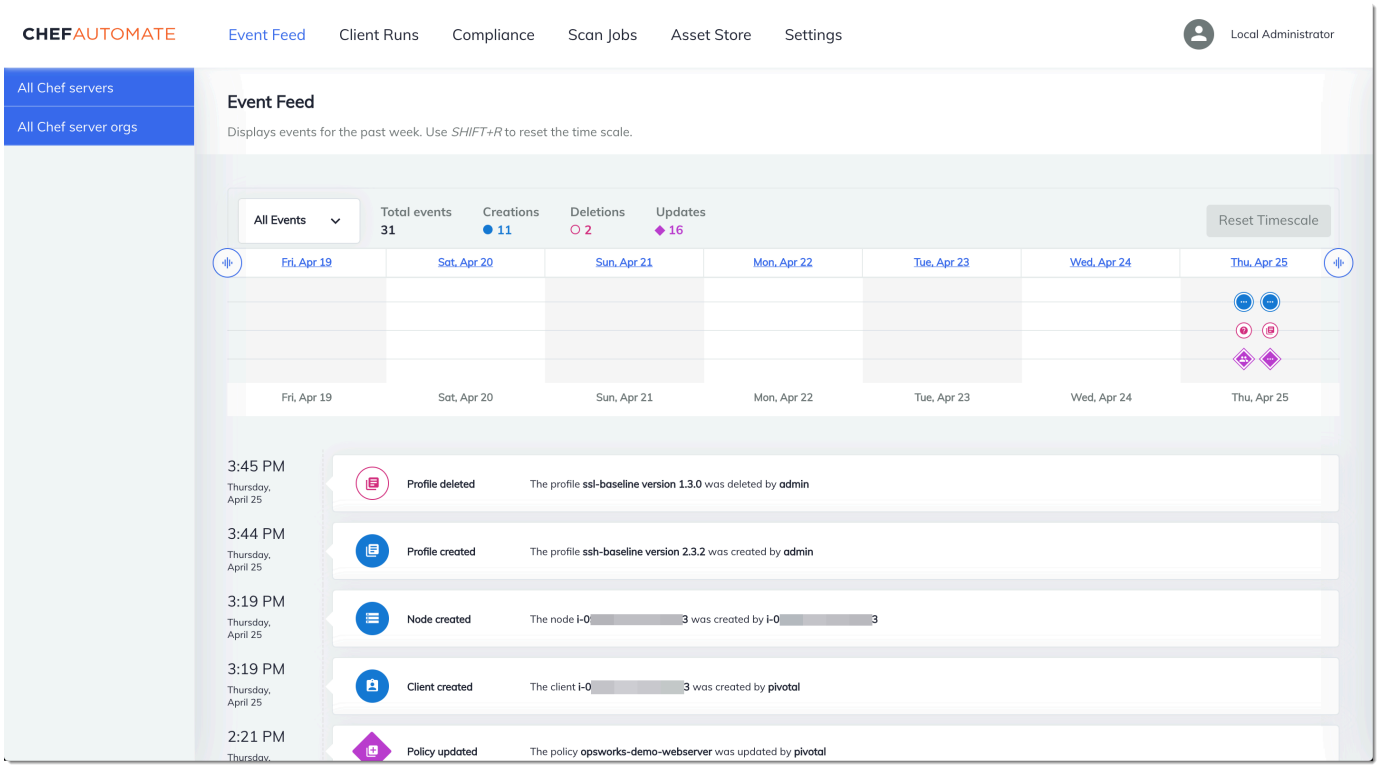
> **Note**
>
> Benutzer von Google Chrome auf den Betriebssystemen Ubuntu und Linux Mint können Schwierigkeiten bei der Anmeldung haben. Wir empfehlen, dass Sie Mozilla Firefox oder andere Browser verwenden, um sich anzumelden und das Chef Automate-Dashboard auf diesen Betriebssystemen zu verwenden. Bei Google Chrome unter Windows oder MacOS wurden keine Probleme berichtet.

So melden Sie sich beim Chef Automate-Dashboard an

1. Entpacken und öffnen Sie die Chef Automate-Anmeldeinformationen, die Sie unter Voraussetzungen heruntergeladen haben. Sie benötigen diese Anmeldeinformationen, um sich anzumelden.
2. Öffnen Sie die Seite Properties (Eigenschaften) für Ihren Chef-Server.
3. Wählen Sie auf der Seite Properties (Eigenschaften) oben rechts Open Chef Automate dashboard (Chef Automate-Dashboard öffnen) aus.
4. Melden Sie sich mit den Anmeldeinformationen aus Schritt 1 an.



5. Im Chef Automate-Dashboard werden detaillierte Informationen zu den installierten Knoten, Rezeptbuch-Ausführungsfortschritten und Ereignissen, Compliance-Ebenen der Knoten und vieles mehr angezeigt. Weitere Informationen zu den Funktionen des Chef Automate-Dashboards und dessen Verwendung finden Sie in der Chef Automate-Dokumentation.



Note

Weitere Informationen zum Ändern des Passworts, mit dem Sie sich beim Chef Automate-Dashboard anmelden, finden Sie unter Zurücksetzen der Anmeldeinformationen für das Chef Automate-Dashboard.

Erstellen Sie einen AWS OpsWorks for Chef Automate Server mit AWS CloudFormation

Important

AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

AWS OpsWorks for Chef Automate ermöglicht es Ihnen, einen Chef Automate-Server in zu betreiben. AWS Sie können in etwa 15 Minuten einen Chef Automate-Server bereitstellen.

AWS OpsWorks for Chef Automate Speichert ab dem 3. Mai 2021 einige Chef Automate-Serverattribute in AWS Secrets Manager. Weitere Informationen finden Sie unter Integration in AWS Secrets Manager.

Die folgende exemplarische Vorgehensweise hilft Ihnen beim Erstellen eines Servers, AWS OpsWorks for Chef Automate indem Sie einen Stack-In AWS CloudFormation erstellen.

Themen

Voraussetzungen

Bevor Sie einen neuen Chef Automate-Server erstellen, erstellen Sie die Ressourcen außerhalb von AWS OpsWorks for Chef Automate , die für den Zugriff auf Ihren Chef-Server und dessen Verwaltung erforderlich sind. Weitere Informationen finden Sie unter Voraussetzungen im Abschnitt "Erste Schritte" dieses Handbuchs.

Im Abschnitt OpsWorks -CM der Vorlagenreferenz für das AWS CloudFormation Benutzerhandbuch finden Sie Informationen zu den unterstützten und erforderlichen Werten in der AWS CloudFormation Vorlage, die Sie zum Erstellen Ihres Servers verwenden.

Wenn Sie einen Server erstellen, der eine benutzerdefinierte Domäne verwendet, benötigen Sie eine benutzerdefinierte Domäne, ein benutzerdefiniertes Zertifikat und eine benutzerdefinierten privaten Schlüssel. Sie müssen Werte für alle drei Parameter in Ihrer AWS CloudFormation Vorlage angeben. Weitere Informationen zu den Anforderungen für die `CustomPrivateKey` Parameter `CustomDomainCustomCertificate`, und finden Sie CreateServerin der AWS OpsWorks CM-API-Referenz.

Erstellen Sie einen Passwortwert für das Engine-Attribut „`CHEF_AUTOMATE_ADMIN_PASSWORD`“. Das Passwort muss eine Länge zwischen 8 und 32 Zeichen haben. Das Passwort kann Buchstaben, Zahlen und Sonderzeichen (`( !/@#$%^+=_)`) enthalten. Es muss mindestens einen Kleinbuchstaben, einen Großbuchstaben, eine Zahl und ein Sonderzeichen enthalten. Sie geben dieses Passwort in Ihrer AWS CloudFormation Vorlage oder als Wert des `CHEF_AUTOMATE_ADMIN_PASSWORD` Parameters an, wenn Sie Ihren Stack erstellen.

Generieren Sie ein Base64-kodiertes RSA-Schlüsselpaar, bevor Sie mit der Erstellung eines Chef Automate-Servers in beginnen. AWS CloudFormation Der öffentliche Schlüssel des Paares ist der Wert von`CHEF_AUTOMATE_PIVOTAL_KEY`, der Chef-spezifisch EngineAttributesaus der API ist. CreateServer Dieser Schlüssel wird als Wert von Parameters in der AWS CloudFormation Konsole oder im create-stack Befehl in der bereitgestellt AWS CLI. Um diesen Schlüssel zu erzeugen, empfehlen wir die folgenden Methoden.

- Auf Linux-basierten Computern können Sie diesen Schlüssel erzeugen, indem Sie den folgenden OpenSSL-Befehl ausführen.

  ```
  openssl genrsa -out pivotal_key_file_name.pem 2048
  ```

  Exportieren Sie dann den öffentlichen RSA-Schlüssel des Paares in eine Datei. Der öffentliche Schlüssel wird zum Wert von `CHEF_AUTOMATE_PIVOTAL_KEY`.

  ```
  openssl rsa -in pivotal_key_file_name.pem -pubout -out public.pem -outform PEM
  ```

- Auf Windows-basierten Computern können Sie das PuTTYgen-Hilfsprogramm verwenden, um ein base64-kodiertes privates RSA-Schlüsselpaar zu generieren. Weitere Informationen finden Sie unter PuTTYgen – Key Generator for PuTTY on Windows auf SSH.com.

Erstellen eines Chef Automate-Servers in AWS CloudFormation

In diesem Abschnitt wird beschrieben, wie Sie mithilfe einer AWS CloudFormation Vorlage einen Stack erstellen, der einen AWS OpsWorks for Chef Automate Server erstellt. Sie können dies tun, indem Sie die AWS CloudFormation Konsole oder die verwenden AWS CLI. Es steht eine AWS CloudFormation Beispielvorlage zur Verfügung, mit der Sie einen AWS OpsWorks for Chef Automate Server-Stack erstellen können. Achten Sie darauf, die Beispielvorlage mit Ihrem eigenen Servernamen, Ihren IAM-Rollen, Ihrem Instanzprofil, Ihrer Serverbeschreibung, der Anzahl der Backup-Aufbewahrungsfristen, den Wartungsoptionen und optionalen Tags zu aktualisieren. Wenn Ihr Server eine benutzerdefinierte Domain verwendet, müssen Sie Werte für die `CustomPrivateKey` Parameter `CustomDomainCustomCertificate`, und in Ihrer AWS CloudFormation Vorlage angeben. Sie können die Attribute `CHEF_AUTOMATE_ADMIN_PASSWORD` und die `CHEF_AUTOMATE_PIVOTAL_KEY` Engine-Attribute und ihre Werte in der AWS CloudFormation Vorlage angeben, oder Sie können nur die Attribute angeben und dann Werte für die Attribute im Assistenten oder create-stack Befehl „Stack AWS CloudFormation erstellen“ angeben. Weitere Informationen zu diesen Attributen finden Sie unter the section called “Erstellen Sie

einen Chef Automate-Server im AWS Management Console" im Abschnitt "Erste Schritte" in diesem Handbuch.

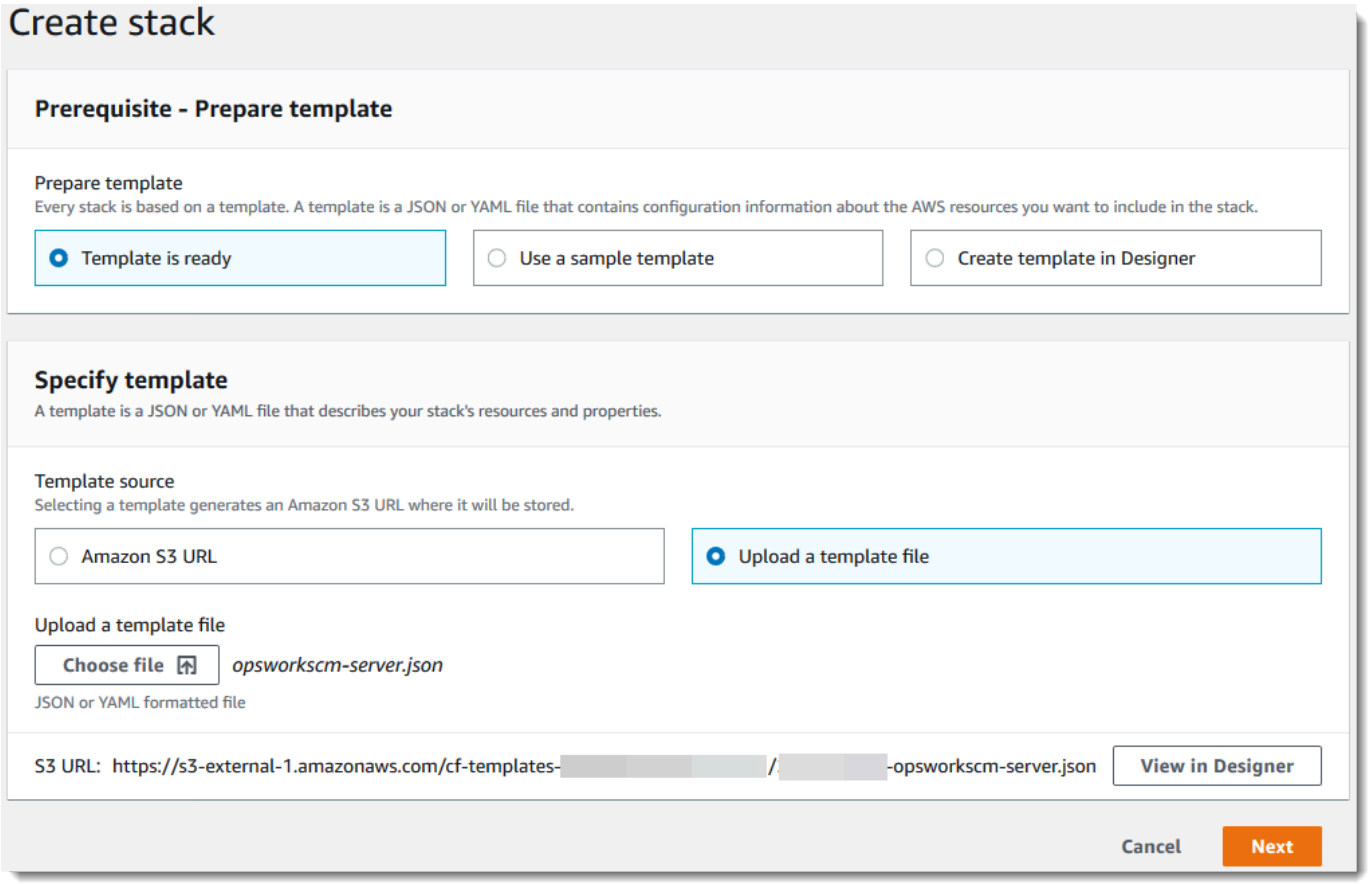
Themen

- Erstellen eines Chef Automate-Servers mit AWS CloudFormation (Konsole)
- Erstellen eines Chef Automate-Servers mit AWS CloudFormation (CLI)

Erstellen eines Chef Automate-Servers mit AWS CloudFormation (Konsole)

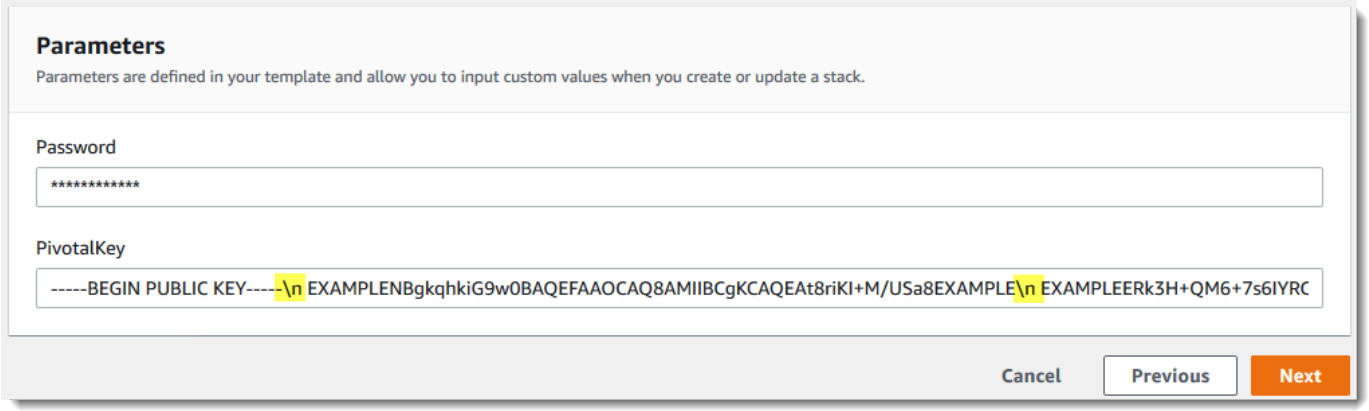
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie auf der AWS CloudFormation Startseite die Option Stack erstellen aus.
3. Wenn Sie die Beispielvorlage verwenden, wählen Sie unter Voraussetzung — AWS CloudFormation Vorlage vorbereiten die Option Vorlage ist bereit aus.
4. Wählen Sie unter Specify template (Vorlage angeben) die Quelle Ihrer Vorlage aus. Wählen Sie für diese exemplarische Vorgehensweise die Option Eine Vorlagendatei hochladen und laden Sie eine AWS CloudFormation Vorlage hoch, die einen Chef Automate-Server erstellt. Suchen Sie nach Ihrer Vorlagendatei und klicken Sie dann auf Next (Weiter).

   Eine AWS CloudFormation Vorlage kann entweder im YAML- oder JSON-Format vorliegen. Es steht Ihnen eine AWS CloudFormation Beispielvorlage zur Verfügung. Achten Sie darauf, die Beispielwerte durch Ihre eigenen zu ersetzen. Sie können den AWS CloudFormation Vorlagendesigner verwenden, um eine neue Vorlage zu erstellen oder eine bestehende zu validieren. Weitere Informationen zu diesem Verfahren finden Sie unter Übersicht über die AWS CloudFormation Designer-Oberfläche im AWS CloudFormation -Benutzerhandbuch.



5. Geben Sie auf der Seite Specify stack details (Stack-Details angeben) einen Namen für den Stack ein. Dies ist nicht dasselbe wie der Name Ihres Servers; es ist nur ein Stack-Name. Fügen Sie im Bereich Parameters (Parameter) die Werte ein, die Sie unter the section called “Voraussetzungen” erstellt haben. Geben Sie unter Password (Passwort) das Passwort ein.

   Fügen Sie den Inhalt der RSA-Schlüsseldatei ein. PivotalKey In der AWS CloudFormation Konsole müssen Sie am Ende jeder Zeile des Schlüsselwerts Zeilenumbrüche (**\n**) hinzufügen, wie im folgenden Screenshot gezeigt. Wählen Sie Weiter aus.



6. Auf der Seite „Stack-Optionen konfigurieren“ können Sie dem Server, den Sie mit dem Stack erstellen, Tags hinzufügen und eine IAM-Rolle für die Erstellung von Ressourcen auswählen, falls Sie noch keine IAM-Rolle zur Verwendung in Ihrer Vorlage angegeben haben. Wenn Sie alle Optionen angegeben haben, wählen Sie Next (Weiter) aus. Weitere Informationen zu erweiterten Optionen wie Rollback-Triggern finden Sie im Benutzerhandbuch unter AWS CloudFormation Stack-Optionen einrichten.AWS CloudFormation
7. Überprüfen Sie auf der Seite Review (Prüfen) Ihre Auswahl. Wenn Sie bereit sind, den Server-Stack zu erstellen, wählen Sie Create stack (Stack erstellen) aus.

   Während Sie darauf warten, den Stack AWS CloudFormation zu erstellen, sehen Sie sich den Status der Stack-Erstellung an. Wenn die Stack-Erstellung fehlschlägt, überprüfen Sie die Fehlermeldungen in der Konsole, die Sie bei der Fehlerbehebung unterstützen. Weitere Informationen zur Fehlerbehebung bei AWS CloudFormation -Stacks finden Sie im Abschnitt zur Fehlerbehebung im AWS CloudFormation -Benutzerhandbuch.

   Wenn die Server-Erstellung abgeschlossen ist, ist Ihr Chef Automate-Server auf der AWS OpsWorks for Chef Automate -Startseite verfügbar und weist den Status online auf. Erstellen Sie auf der Seite "Properties (Eigenschaften)" des Servers ein neues Starter Kit und die Anmeldeinformationen für das Chef Automate-Dashboard. Nachdem sich der Server online befindet, ist das Chef Automate-Dashboard auf der Server-Domäne mit einer URL mit folgendem Format verfügbar: `https://your_server_name-randomID.region.opsworks-cm.io`.

   > **Note**
   >
   > Wenn Sie eine benutzerdefinierte Domäne, ein Zertifikat und einen privaten Schlüssel für Ihren Server angegeben haben, erstellen Sie im DNS-Verwaltungstool Ihres Unternehmens einen CNAME-Eintrag, der Ihre benutzerdefinierte Domain dem Endpunkt zuordnet, der AWS OpsWorks for Chef Automate automatisch für den Server generiert wurde. Sie können den Server erst verwalten oder erst dann eine Verbindung mit dem Chef Automate-Dashboard für den Server herstellen, nachdem Sie den generierten Endpunkt Ihrem benutzerdefinierten Domänenwert zugeordnet haben.
   > Um den generierten Endpunktwert abzurufen, führen Sie den folgenden AWS CLI Befehl aus, nachdem Ihr Server online ist:
   >
   > ```
   > aws opsworks describe-servers --server-name server_name
   > ```

Erstellen eines Chef Automate-Servers mit AWS CloudFormation (CLI)

Wenn auf Ihrem lokalen Computer das noch nicht ausgeführt wird AWS CLI, laden Sie es herunter und installieren Sie es, AWS CLI indem Sie den Installationsanweisungen im AWS-Benutzerhandbuch für die Befehlszeilenschnittstelle folgen. In diesem Abschnitt werden nicht alle Parameter beschrieben, die Sie mit dem Befehl create-stack verwenden können. Weitere Informationen zu den create-stack-Parametern finden Sie unter create-stack in der AWS CLI - Referenz.

1. Stellen Sie sicher, dass Sie die Voraussetzungen für das Erstellen eines AWS OpsWorks for Chef Automate -Servers erfüllt haben.
2. Erstellen Sie eine Servicerolle und ein Instanzprofil. AWS OpsWorks stellt eine AWS CloudFormation Vorlage bereit, mit der Sie beide erstellen können. Führen Sie den folgenden AWS CLI Befehl aus, um einen AWS CloudFormation Stack zu erstellen, der die Servicerolle und das Instanzprofil für Sie erstellt.

   ```
   aws cloudformation create-stack --stack-name OpsWorksCMRoles --template-url
    https://s3.amazonaws.com/opsworks-cm-us-east-1-prod-default-assets/misc/opsworks-
   cm-roles.yaml --capabilities CAPABILITY_NAMED_IAM
   ```

   Suchen Sie nach AWS CloudFormation Abschluss der Erstellung des Stacks die ARNs der Servicerollen in Ihrem Konto und kopieren Sie sie.

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

   Suchen Sie in den Ergebnissen des Befehls `list-roles` nach den Einträgen der Servicerolle und des Instance-Profils. Diese sehen etwa wie folgt aus. Notieren Sie sich die ARNs der Servicerolle und des Instanzprofils und fügen Sie sie der AWS CloudFormation Vorlage hinzu, mit der Sie Ihren Server-Stack erstellen.

   ```
   {
       "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. Erstellen Sie den AWS OpsWorks for Chef Automate Server, indem Sie den create-stack Befehl erneut ausführen.

   - Ersetzen Sie *`stack_name`* durch den Namen Ihres Stacks. Dies ist der Name des AWS CloudFormation Stacks, nicht Ihr Chef Automate-Server. Der Chef Automate-Servername ist der Wert von `ServerName` in Ihrer AWS CloudFormation Vorlage.
   - Ersetzen Sie *`template`* durch den Pfad zu Ihrer Vorlagendatei und die Erweiterung *`yaml or json`* mit `.yaml` bzw. `.json` (wie zutreffend).
   - Die Werte für `--parameters` entsprechen EngineAttributesder CreateServerAPI. Für Chef lauten die zur Erstellung eines Servers genutzten, vom Benutzer angegebenen Engine-Attribute „`CHEF_AUTOMATE_PIVOTAL_KEY`“, ein base64-kodierter öffentlicher RSA-Schlüssel,

den Sie mit Dienstprogrammen wie unter „the section called “Voraussetzungen”“ beschrieben erstellen, sowie „CHEF_AUTOMATE_ADMIN_PASSWORD“, ein Passwort mit 8 bis 32 Zeichen, das Sie erstellen. Weitere Informationen zu CHEF_AUTOMATE_ADMIN_PASSWORDfinden Sie unter Erstellen Sie einen Chef Automate-Server mit dem AWS CLI. Sie können einen Zeiger zu der PEM-Datei angeben, die Ihren pivotalen Schlüssel als Wert des Parameters PivotalKey enthält, wie im Beispiel gezeigt. Wenn die Werte für CHEF_AUTOMATE_ADMIN_PASSWORD und nicht in Ihrer Vorlage angegeben CHEF_AUTOMATE_PIVOTAL_KEY sind, müssen Sie die Werte in Ihrem AWS CLI Befehl angeben.

```
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"
```

Nachfolgend sehen Sie ein Beispiel mit Beispielwerten für die Attribute „CHEF_AUTOMATE_ADMIN_PASSWORD“ und „CHEF_AUTOMATE_PIVOTAL_KEY“. Führen Sie einen ähnlichen Befehl aus, wenn Sie in Ihrer AWS CloudFormation Vorlage keine Werte für diese Attribute angegeben haben.

```
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. Wenn die Stack-Erstellung abgeschlossen ist, öffnen Sie die Eigenschaftenseite für den neuen Server in der AWS OpsWorks for Chef Automate Konsole und laden Sie ein Starterkit herunter. Wenn Sie ein neues Starter Kit herunterladen, wird das Administratorpasswort des Chef Automate-Dashboards zurückgesetzt.
5. Wenn Ihr Server eine benutzerdefinierte Domäne, ein Zertifikat und einen privaten Schlüssel verwendet, führen Sie die Schritte für die Konfiguration von `knife.rb` in (Optional) Konfigurieren von `knife` zur Verwendung mit einer benutzerdefinierten Domäne aus und fahren Sie mit Schritt 7 fort.

   Wenn Sie keine benutzerdefinierte Domain verwenden, laden Sie das Zertifikat der Root-Zertifizierungsstelle (CA) vom folgenden Amazon S3 S3-Bucket-Speicherort herunter:https://s3.amazonaws.com/opsworks-cm-us-east-1-prod-default-assets/misc/opsworks-cm-ca-2020-

root.pem. Speichern Sie die Zertifikatsdatei an einem sicheren, aber praktischen Speicherort. Dieses Zertifikat ist erforderlich, um `knife.rb` im nächsten Schritt zu konfigurieren.

6. Um die `knife`-Befehle auf dem neuen Server zu verwenden, aktualisieren Sie die Einstellungen der Chef-Konfigurationsdatei `knife.rb`. Ein Beispiel für die Datei `knife.rb` ist im Starter Kit enthalten. Das folgende Beispiel zeigt, wie Sie `knife.rb` auf einem Server einrichten, der keine benutzerdefinierte Domäne verwendet. Wenn Sie eine benutzerdefinierte Domäne verwenden, finden Sie unter (Optional) Konfigurieren von `knife` zur Verwendung mit einer benutzerdefinierten Domäne `knife`-Konfigurationsanweisungen.

   - Ersetzen Sie *`ENDPOINT`* mit dem Endpunktwert des Servers. Dies ist Teil der Ausgabe des Stack-Erstellungsvorgangs. Sie erhalten den Endpunkt, indem Sie den folgenden Befehl ausführen.

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

   - Ersetzen Sie *`key_pair_file.pem`* in der `client_key`-Konfiguration durch den Namen der PEM-Datei, die den `CHEF_AUTOMATE_PIVOTAL_KEY` enthält, den Sie bei der Erstellung Ihres Servers verwendet haben.

     ```
     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. Fahren Sie nach Abschluss der Servererstellung mit the section called “Konfiguration abschließen und Rezeptbuch hochladen” fort. Wenn die Stack-Erstellung fehlschlägt, überprüfen Sie die Fehlermeldungen in der Konsole, die Sie bei der Fehlerbehebung unterstützen. Weitere Informationen zur Behebung von Fehlern in AWS CloudFormation Stacks finden Sie unter Problembehebung im AWS CloudFormation Benutzerhandbuch.

Einen AWS OpsWorks for Chef Automate Server für die Verwendung einer benutzerdefinierten Domain aktualisieren

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Lebensende erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

In diesem Abschnitt wird beschrieben, wie Sie einen vorhandenen AWS OpsWorks for Chef Automate Server für die Verwendung einer benutzerdefinierten Domain und eines benutzerdefinierten Zertifikats aktualisieren, indem Sie ein Backup des Servers verwenden, um einen neuen Server zu erstellen. Im Wesentlichen kopieren Sie einen vorhandenen AWS OpsWorks for Chef Automate 2.0-Server, indem Sie einen neuen Server aus einem Backup erstellen und den neuen Server dann so konfigurieren, dass er eine benutzerdefinierte Domäne, ein benutzerdefiniertes Zertifikat und einen privaten Schlüssel verwendet.

Themen

Voraussetzungen

Im Folgenden sind die Anforderungen für die Aktualisierung eines vorhandenen AWS OpsWorks for Chef Automate Servers für die Verwendung einer benutzerdefinierten Domäne und eines benutzerdefinierten Zertifikats aufgeführt.

- Auf dem Server, den Sie aktualisieren (oder kopieren) möchten, muss Chef Automate 2.0 ausgeführt werden.
- Legen Sie fest, welche Sicherung Sie zum Erstellen eines neuen Servers verwenden möchten. Sie müssen mindestens eine Sicherung des Servers zur Verfügung haben, den Sie aktualisieren

möchten. Weitere Informationen zu Backups in AWS OpsWorks for Chef Automate finden Sie unterEinen AWS OpsWorks for Chef Automate Server sichern.

- Bereiten Sie die Servicerollen- und Instance-Profil-ARNs vor, mit denen Sie den vorhandenen Server erstellt haben, der die Quelle Ihrer Sicherung ist.
- Stellen Sie sicher, dass Sie die neueste Version von AWS CLI ausführen. Weitere Informationen zur Aktualisierung Ihrer AWS CLI Tools finden Sie unter Installation von AWS CLI im AWS-Benutzerhandbuch für die Befehlszeilenschnittstelle.

Einschränkungen

Wenn Sie einen vorhandenen Server aktualisieren, indem Sie einen neuen Server aus einem Backup erstellen, kann der neue Server nicht exakt mit dem vorhandenen AWS OpsWorks for Chef Automate Server identisch sein.

- Sie können dieses Verfahren nur mit dem AWS CLI oder einem der AWS SDKs abschließen. Sie können mit der AWS Management Console keinen neuen Server aus einer Sicherung erstellen.
- Der neue Server kann nicht denselben Namen wie der vorhandene Server in einem Konto und in einer AWS-Region verwenden. Der Name muss sich von dem vorhandenen Server unterscheiden, den Sie als Quelle der Sicherung verwendet haben.
- Knoten, die an den vorhandenen Server angeschlossen wurden, werden nicht vom neuen Server verwaltet. Sie müssen einen der folgenden Schritte ausführen.
  - Fügen Sie verschiedene Knoten hinzu, da Knoten von nicht von mehr als einem Chef Automate-Server verwaltet werden können.
  - Migrieren Sie die Knoten vom vorhandenen Server (der Quelle der Sicherung) auf den neuen Server und den neuen benutzerdefinierten Domänenendpunkt. Weitere Informationen zum Migrieren von Knoten finden Sie in der Chef-Dokumentation.

Aktualisieren eines Servers zur Verwendung einer benutzerdefinierten Domäne

Um einen vorhandenen Chef Automate 2.0-Server zu aktualisieren, erstellen Sie eine Kopie davon, indem Sie den Befehl `create-server` ausführen und Parameter hinzufügen, um eine Sicherung, eine benutzerdefinierte Domäne, ein benutzerdefiniertes Zertifikat und einen benutzerdefinierten privaten Schlüssel anzugeben.

1. Wenn Sie in Ihrem `create-server`-Befehl keine Servicerollen- oder Instance-Profil-ARNs zur Verfügung haben, führen Sie die Schritte 1-5 in Erstellen Sie einen Chef Automate-Server mit dem AWS CLI aus, um eine Servicerolle und ein Instance-Profil zu erstellen, die Sie verwenden können.
2. Wenn Sie dies noch nicht getan haben, suchen Sie die Sicherung des vorhandenen Chef Automate 2.0-Servers, auf dem Sie einen neuen Server mit einer benutzerdefinierten Domäne erstellen möchten. Führen Sie den folgenden Befehl aus, um Informationen zu allen AWS OpsWorks for Chef Automate Backups in Ihrem Konto und in einer Region anzuzeigen. Notieren Sie sich die ID der Sicherung, die Sie verwenden möchten.

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

3. Erstellen Sie den AWS OpsWorks for Chef Automate Server, indem Sie den `create-server` Befehl ausführen.

   - Der Wert `--engine` ist `ChefAutomate`, `--engine-model` ist `Single` und `--engine-version` ist 12.
   - Der Servername muss innerhalb Ihres AWS Kontos in jeder Region eindeutig sein. Servernamen müssen mit einem Buchstaben beginnen. Danach können Buchstaben, Zahlen und Bindestriche (-) verwendet werden, insgesamt höchstens 40 Zeichen.
   - Verwenden Sie den Instance-Profil-ARN und den Servicerollen-ARN aus Schritt 1.
   - Gültige Instance-Typen sind `m5.large`, `r5.xlarge` und `r5.2xlarge`. Weitere Informationen zu den Spezifikationen dieser Instance-Typen finden Sie unter Instance-Typen im Amazon EC2 EC2-Benutzerhandbuch.
   - Der Parameter `--engine-attributes` ist optional. Wenn Sie nicht einen oder beide Werte festlegen, werden die Werte bei der Servererstellung für Sie generiert. Wenn Sie `--engine-attributes` hinzufügen, geben Sie entweder den Wert `CHEF_AUTOMATE_PIVOTAL_KEY`, den Sie in Schritt 2 generiert haben, ein `CHEF_AUTOMATE_ADMIN_PASSWORD` oder beides an.

     Wenn Sie keinen Wert für `CHEF_AUTOMATE_ADMIN_PASSWORD` festlegen, wird ein Passwort für Sie generiert und in der Antwort des Befehls `create-server` zurückgegeben. Sie können auch das Starter Kit in der Konsole erneut herunterladen, um dieses Passwort erneut zu generieren. Das Passwort muss eine Länge zwischen 8 und 32 Zeichen haben. Das Passwort kann Buchstaben, Zahlen und Sonderzeichen (`!/@#$%^+=_`) enthalten. Es muss mindestens einen Kleinbuchstaben, einen Großbuchstaben, eine Zahl und ein Sonderzeichen enthalten.

- Ein SSH-Schlüsselpaar ist optional. Es kann Ihnen dabei helfen, sich mit dem Chef Automate-Server zu verbinden, wenn Sie das Administratorpasswort des Chef Automate-Dashboards zurücksetzen müssen. Weitere Informationen zum Erstellen eines SSH-Schlüsselpaars finden Sie unter Amazon EC2 EC2-Schlüsselpaare im Amazon EC2 EC2-Benutzerhandbuch.
- Um eine benutzerdefinierte Domäne zu verwenden, fügen Sie dem Befehl die folgenden Parameter hinzu. Andernfalls generiert der Erstellungsprozess des Chef Automate-Servers automatisch einen Endpunkt für Sie. Alle drei Parameter sind erforderlich, um eine benutzerdefinierte Domäne zu konfigurieren. Informationen zu zusätzlichen Anforderungen für die Verwendung dieser Parameter finden Sie CreateServerin der AWS OpsWorks CM-API-Referenz.
  - `--custom-domain` – Ein optionaler öffentlicher Endpunkt eines Servers, z. B. `https://aws.my-company.com`.
  - `--custom-certificate` – Ein PEM-formatiertes HTTPS-Zertifikat. Der Wert kann ein einzelnes, selbstsigniertes Zertifikat oder eine Zertifikatkette sein.
  - `--custom-private-key` – Ein privater Schlüssel im PEM-Format für die Verbindung mit dem Server mithilfe von HTTPS. Der private Schlüssel darf nicht verschlüsselt sein; er kann nicht durch ein Passwort oder eine Passphrase geschützt werden.
- Es ist eine wöchentliche Systemwartung erforderlich. Gültige Werte müssen im folgenden Format angegeben werden: `DDD:HH:MM`. Die angegebene Uhrzeit entspricht der Zeitzone UTC (Coordinated Universal Time). Wenn Sie für `--preferred-maintenance-window` keinen Wert angeben, wird ein zufälliger Standardwert mit einem einstündigen Zeitraum an einem Dienstag, Mittwoch oder Freitag festgelegt.
- Gültige Werte für `--preferred-backup-window` müssen in einem der folgenden Formate angegeben werden: `HH:MM` für tägliche Sicherungen oder `DDD:HH:MM` für wöchentliche Sicherungen. Die angegebene Uhrzeit entspricht der Zeitzone UTC. Standardmäßig wird ein zufälliger täglicher Startzeitpunkt festgelegt. Wenn Sie automatische Sicherungen deaktivieren möchten, verwenden Sie stattdessen den Parameter `--disable-automated-backup`.
- Geben Sie für `--security-group-ids` eine oder mehrere Sicherheitsgruppen-IDs, durch Kommata getrennt, ein.
- Geben Sie für `--subnet-ids` eine Subnetz-ID ein.
- Geben Sie für `--backup-id` die ID der Sicherung ein, die Sie in Schritt 2 kopiert haben.

```
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
```

Im folgenden Beispiel wird ein Chef Automate-Server erstellt, der eine benutzerdefinierte Domäne verwendet.

```
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 benötigt etwa 15 Minuten, um einen neuen Server zu erstellen. Kopieren Sie in der Ausgabe des Befehls `create-server` den Wert des Attributs `Endpoint`. Im Folgenden wird ein Beispiel gezeigt.

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

Sie sollten die Ausgabe des Befehls `create-server` nicht verwerfen oder die Shell-Sitzung beenden, da die Ausgabe wichtige Informationen enthalten kann, die nicht wiederhergestellt

werden können. Um Passwörter und das Starter Kit aus den Ergebnissen von `create-server` zu extrahieren, fahren Sie mit dem nächsten Schritt fort.

5. Wenn Sie sich dafür entschieden haben, einen Schlüssel und ein Passwort für Sie AWS OpsWorks for Chef Automate generieren zu lassen, können Sie diese mithilfe eines JSON-Prozessors wie jq in verwendbaren Formaten aus den `create-server` Ergebnissen extrahieren. Nachdem Sie jq installiert haben, können Sie die folgenden Befehle ausführen, um den pivotalen Schlüssel, das Administratorpasswort für das Chef Automate-Dashboard und das Starter Kit zu extrahieren. Wenn Sie in Schritt 3 keinen eigenen pivotalen Schlüssel und kein eigenes Passwort angegeben haben, speichern Sie den extrahierten pivotalen Schlüssel und das extrahierte Administratorpasswort an einem sicheren Speicherort.

```
#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. Falls Sie das Starterkit nicht aus den `create-server` Befehlsergebnissen extrahiert haben, können Sie optional ein neues Starterkit von der Eigenschaftenseite des Servers in der AWS OpsWorks for Chef Automate Konsole herunterladen. Wenn Sie ein neues Starter Kit herunterladen, wird das Administratorpasswort des Chef Automate-Dashboards zurückgesetzt.
7. Erstellen Sie einen CNAME-Eintrag im DNS-Verwaltungstool Ihres Unternehmens, um Ihre benutzerdefinierte Domain auf den AWS OpsWorks for Chef Automate Endpunkt zu verweisen, den Sie in Schritt 4 kopiert haben. Sie können den Server erst erreichen oder sich erst dann anmelden, nachdem Sie diesen Schritt ausgeführt haben.
8. Fahren Sie nach Abschluss der Servererstellung mit the section called “Konfiguration abschließen und Rezeptbuch hochladen” fort.

Weitere Informationen finden Sie unter:

- Erstellen Sie einen Chef Automate-Server mit dem AWS CLI

- Einen AWS OpsWorks for Chef Automate Server aus einem Backup wiederherstellen
- CreateServerin der AWS OpsWorks CM-API-Referenz
- create-server in der AWS CLI Befehlsreferenz

Regenerieren Sie das Starterkit für einen Server AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

Das Starterkit für AWS OpsWorks for Chef Automate enthält eine README-Datei mit Beispielen, eine `knife.rb` Konfigurationsdatei und einen privaten Schlüssel für den primären oder zentralen Benutzer. Jedes Mal, wenn Sie das Starterkit herunterladen, wird ein neues key pair generiert — und der alte Schlüssel wird zurückgesetzt. Sie können das Starterkit für einen AWS OpsWorks for Chef Automate Server auf zwei Arten neu generieren:

- In der AWS OpsWorks Konsole im Menü Aktionen der Detailseite für einen AWS OpsWorks for Chef Automate Server. Sie werden aufgefordert zu bestätigen, ob Sie den alten Pivotalschlüssel regenerieren und zurücksetzen möchten.
- Durch das Ausführen von Befehlen in der. AWS CLI

Weitere Informationen zur Verwendung des Starterkits finden Sie unterKonfigurieren des Chef-Servers mit dem Starter Kit.

Regenerieren Sie das AWS OpsWorks for Chef Automate Starterkit mit dem AWS CLI

> **Note**
>
> Wenn Sie das Starterkit regenerieren, regenerieren und setzen Sie auch das Authentifizierungsschlüsselpaar für Ihren Chef Automate-Server zurück und löschen das aktuelle key pair.

Regenerieren Sie das Starterkit, indem Sie den Befehl ausführen. update-server-engine-attributes Führen Sie in einer AWS CLI Sitzung den folgenden Befehl aus. Geben Sie Ihren Servernamen als Wert von an`--server-name`. Um einen eigenen öffentlichen Schlüssel als Wert von festzulegen`CHEF_AUTOMATE_PIVOTAL_KEY`, geben Sie den Wert des öffentlichen Schlüssels in an`--attribute-value`. Andernfalls setzen Sie ihn `--attribute-value` auf Null.

```
aws opsworks-cm update-server-engine-attributes \
   --server-name server_name \
   --attribute-name "CHEF_AUTOMATE_PIVOTAL_KEY" \
   --attribute-value your_public_key
```

Der folgende Befehl ist ein Beispiel, das einen öffentlichen Schlüsselwert angibt, den der Serveradministrator verwenden möchte.

```
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-----"
```

Der folgende Befehl ist ein Beispiel, mit dem der öffentliche Schlüssel AWS OpsWorks for Chef Automate neu generiert werden kann.

```
aws opsworks-cm update-server-engine-attributes \
   --server-name your-test-server \
   --attribute-name "CHEF_AUTOMATE_PIVOTAL_KEY" \
   --attribute-value null
```

Die Ausgabe dieses Befehls besteht aus Informationen über den Server und einer Base64-codierten ZIP-Datei. Die ZIP-Datei enthält ein Chef-Starterkit, das eine README-Datei, eine Konfigurationsdatei und den erforderlichen privaten RSA-Schlüssel enthält. Speichern Sie diese Datei, entpacken Sie sie und wechseln Sie dann in das Verzeichnis, in das Sie den Dateiinhalt entpackt haben. In diesem Verzeichnis können Sie Befehle ausführen. `knife`

Mit Tags auf AWS OpsWorks for Chef Automate Ressourcen arbeiten

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Lebensende erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

Tags sind Wörter oder Ausdrücke, die in Form von Metadaten zum Identifizieren und Organisieren Ihrer AWS-Ressourcen verwendet werden. In AWS OpsWorks for Chef Automate kann eine Ressource bis zu 50 vom Benutzer angewendete Tags haben. Jedes Tag besteht aus einem Schlüssel und einem einzelnen optionalen Wert. Sie können Tags auf die folgenden Ressourcen in AWS OpsWorks for Chef Automate anwenden:

- AWS OpsWorks for Chef Automate Server
- Backups von AWS OpsWorks for Chef Automate Servern

Mithilfe von Tags auf AWS Ressourcen können Sie Kosten verfolgen, den Zugriff auf Ressourcen kontrollieren, Ressourcen zur Automatisierung von Aufgaben gruppieren oder Ressourcen nach Zweck oder Lebenszyklusphase organisieren. Weitere Informationen zu den Vorteilen von Tags finden Sie unter AWS-Tagging-Strategien in AWS Answers und unter Verwendung von Kostenzuweisungs-Tags im AWS Billing and Cost Management -Benutzerhandbuch.

Um mithilfe von Tags den Zugriff auf AWS OpsWorks for Chef Automate Server oder Backups zu kontrollieren, erstellen oder bearbeiten Sie Richtlinienerklärungen in AWS Identity and Access Management (IAM). Weitere Informationen finden Sie unter Steuern des Zugriffs auf

AWS -Ressourcen mithilfe von Ressourcen-Tags im AWS Identity and Access Management - Benutzerhandbuch.

Wenn Sie Tags auf einen AWS OpsWorks for Chef Automate Server anwenden, werden die Tags auch auf die Backups des Servers, den Amazon S3 S3-Bucket, in dem die Backups gespeichert sind, die Amazon EC2 EC2-Instance des Servers, die auf dem Server gespeichert sind AWS Secrets Manager, und die vom Server verwendete Elastic IP-Adresse angewendet. Tags werden nicht an den AWS CloudFormation Stack weitergegeben, der zur Erstellung Ihres Servers AWS OpsWorks verwendet wird.

Themen

So funktionieren Tags in AWS OpsWorks for Chef Automate

In dieser Version können Sie Tags hinzufügen und verwalten, indem Sie die AWS OpsWorks - CM-API oder die AWS Management Console verwenden. AWS OpsWorks CM versucht auch, Tags, die Sie einem Server hinzufügen, zu den AWS Ressourcen hinzuzufügen, die dem Server zugeordnet sind, einschließlich der EC2-Instance, Secrets in Secrets Manager, Elastic IP-Adresse, Sicherheitsgruppe, S3-Bucket und Backups. In der folgenden Tabelle finden Sie eine Übersicht darüber, wie Sie Tags in AWS OpsWorks for Chef Automate hinzufügen und verwalten.

Aktion	Was zu verwenden ist
Fügen Sie einem neuen AWS OpsWorks for Chef Automate Server oder einem Backup, das Sie manuell erstellen, Tags hinzu.	• Wählen Sie Create Chef Automate Server (Chef Automate-Server erstellen) aus und fügen Sie Tags auf der Seite Configure advanced settings (Erweiterte Einstellungen konfigurieren) hinzu. • Wählen Sie Backup erstellen auf der Seite Backups für einen vorhandenen Server und fügen Sie Tags auf der Seite Backup Ihres Chef Automate 2-Servers erstellen hinzu.

Aktion	Was zu verwenden ist
	• Fügen Sie den Befehlen „CreateServer“ oder „CreateBackup“ einen Tags-Parameter hinzu.
Anzeigen von Tags auf einer Ressource.	• Wählen Sie auf der Detailseite für Ihren Server im Navigationsbereich die Option Tags aus. • Wählen Sie auf der Seite Backups (Sicherungen) für Ihren Server eine Sicherung und dann Edit backup (Sicherung bearbeiten) aus. • Führen Sie den Befehl ListTagsForResource aus.
Fügen Sie einem vorhandenen AWS OpsWorks for Chef Automate Server oder einem Backup Tags hinzu, unabhängig davon, ob das Backup manuell oder automatisch erstellt wurde.	• Wählen Sie auf der Detailseite für Ihren Server im Navigationsbereich die Option Tags und dann Edit (Bearbeiten)aus. • Wählen Sie auf der Seite Backups (Sicherungen) für Ihren Server eine Sicherung und dann Edit backup (Sicherung bearbeiten) aus. • Führen Sie den Befehl TagResource aus.

Aktion	Was zu verwenden ist
Löschen Sie Tags von einer Ressource.	• Wählen Sie auf der Detailseite für Ihren Server im Navigationsbereich die Option Tags und dann Edit (Bearbeiten)aus. Wählen Sie das X neben den Tags aus, die Sie löschen möchten. • Wählen Sie auf der Seite Backups (Sicherungen) für Ihren Server eine Sicherung und dann Edit backup (Sicherung bearbeiten) aus. Wählen Sie das X neben den Tags aus, die Sie löschen möchten. • Führen Sie den Befehl `UntagResource` aus.

`DescribeServers`- und `DescribeBackups`-Antworten enthalten keine Tag-Informationen. Verwenden Sie die `ListTagsForResource`-API, um Tags anzuzeigen.

Hinzufügen und Verwalten von Tags in AWS OpsWorks for Chef Automate (Konsole)

Die Prozeduren in diesem Abschnitt werden in der AWS Management Console durchgeführt.

Wenn Sie Tags hinzufügen, darf ein Tag-Schlüssel nicht leer sein. Der Schlüssel darf maximal 127 Zeichen lang sein und nur Unicode-Buchstaben, Ziffern oder Trennzeichen oder die folgenden Sonderzeichen enthalten: `+ - = . _ : / @`-Tag-Werte sind optional. Sie können einen Tag hinzufügen, der einen Schlüssel, aber keine Werte enthält. Der Wert darf maximal 255 Zeichen lang sein und können nur Unicode-Buchstaben, Ziffern oder Trennzeichen oder die folgenden Sonderzeichen enthalten: `+ - = . _ : / @`.

Themen

- Löschen von Tags aus einem Server (Konsole)
- Löschen von Tags aus einer Sicherung (Konsole)

Hinzufügen von Tags zu einem neuen AWS OpsWorks for Chef Automate Server (Konsole)

1. Stellen Sie sicher, dass alle Voraussetzungen für die Erstellung eines AWS OpsWorks for Chef Automate Servers erfüllt sind.
2. Führen Sie die Schritte 1-10 in Erstellen eines Chef Automate-Servers aus.
3. Nachdem Sie automatische Sicherungseinstellungen festgelegt haben, fügen Sie Tags im Bereich Tags der Seite Configure advanced settings (Erweiterte Einstellungen konfigurieren) hinzu. Sie können maximal 50 Tags hinzufügen. Wenn Sie alle Tags hinzugefügt haben, wählen Sie Next aus.
4. Fahren Sie mit Schritt 13 von Erstellen eines Chef Automate-Servers fort und überprüfen Sie die Einstellungen, die Sie für den neuen Server ausgewählt haben.

Hinzufügen von Tags zu einer neuen Sicherung (Konsole)

1. Wählen Sie auf der AWS OpsWorks for Chef Automate Startseite einen vorhandenen Chef Automate-Server aus.
2. Wählen Sie auf der Detailseite des Servers im Navigationsbereich Backups (Sicherungen) aus.
3. Wählen Sie auf der Seite Backups (Sicherungen) die Option Create Backup (Sicherung erstellen) aus.
4. Fügen Sie Tags hinzu. Wählen Sie Create (Erstellen) aus, sobald Sie mit dem Hinzufügen von Tags fertig sind.

Hinzufügen oder Anzeigen von Tags auf einem vorhandenen Server (Konsole)

1. Wählen Sie auf der AWS OpsWorks for Chef Automate Startseite einen vorhandenen Chef Automate-Server aus, um dessen Detailseite zu öffnen.
2. Wählen Sie im Navigationsbereich Tags oder unten auf der Detailseite die Option View all tags (Alle Tags anzeigen) aus.
3. Wählen Sie auf der Seite Tags die Option Edit (Bearbeiten) aus.

4. Tags auf dem Server hinzufügen oder bearbeiten. Wählen Sie Save (Speichern) aus, wenn Sie fertig sind.

> **Note**
>
> Beachten Sie, dass das Ändern von Tags auf Ihrem Chef Automate-Server auch Tags für Ressourcen ändert, die dem Server zugewiesen sind, z, B. die EC2-Instance, die Elastic IP-Adresse, die Sicherheitsgruppe, den S3-Bucket und Sicherungen.

Hinzufügen oder Anzeigen von Tags in einer vorhandenen Sicherung (Konsole)

1. Wählen Sie auf der AWS OpsWorks for Chef Automate Startseite einen vorhandenen Chef Automate-Server aus, um dessen Detailseite zu öffnen.
2. Wählen Sie im Navigationsbereich Backups (Sicherungen) oder im Bereich Recent backups (Zuletzt verwendete Sicherungen) auf der Detailseite die Option View all backups (Alle Sicherungen anzeigen) aus.
3. Wählen Sie auf der Seite Backups (Sicherungen) eine zu verwaltende Sicherung aus, und wählen Sie dann Edit Backups (Sicherung bearbeiten) aus.
4. Tags in der Sicherung hinzufügen oder bearbeiten. Wählen Sie Update (Aktualisieren) aus, wenn Sie fertig sind.

Löschen von Tags aus einem Server (Konsole)

1. Wählen Sie auf der AWS OpsWorks for Chef Automate Startseite einen vorhandenen Chef Automate-Server aus, um dessen Detailseite zu öffnen.
2. Wählen Sie im Navigationsbereich Tags oder unten auf der Detailseite die Option View all tags (Alle Tags anzeigen) aus.
3. Wählen Sie auf der Seite Tags die Option Edit (Bearbeiten) aus.
4. Wählen Sie das X neben einem Tag aus, um das Tag zu löschen. Wählen Sie Save (Speichern) aus, wenn Sie fertig sind.

> **Note**
>
> Beachten Sie, dass das Ändern von Tags auf Ihrem Chef Automate-Server auch Tags für Ressourcen ändert, die dem Server zugewiesen sind, z, B. die EC2-Instance, die Elastic IP-Adresse, die Sicherheitsgruppe, den S3-Bucket und Sicherungen.

Löschen von Tags aus einer Sicherung (Konsole)

1. Wählen Sie auf der AWS OpsWorks for Chef Automate Startseite einen vorhandenen Chef Automate-Server aus, um dessen Detailseite zu öffnen.
2. Wählen Sie im Navigationsbereich Backups (Sicherungen) oder im Bereich Recent backups (Zuletzt verwendete Sicherungen) auf der Detailseite die Option View all backups (Alle Sicherungen anzeigen) aus.
3. Wählen Sie auf der Seite Backups (Sicherungen) eine zu verwaltende Sicherung aus, und wählen Sie dann Edit Backups (Sicherung bearbeiten) aus.
4. Wählen Sie das X neben einem Tag aus, um das Tag zu löschen. Wählen Sie Update (Aktualisieren) aus, wenn Sie fertig sind.

Hinzufügen und Verwalten von Tags in AWS OpsWorks for Chef Automate (CLI)

Die Prozeduren in diesem Abschnitt werden in der AWS CLI durchgeführt. Stellen Sie sicher, dass Sie die neueste Version von ausführen, AWS CLI bevor Sie mit der Arbeit mit Tags beginnen. Weitere Informationen zur Installation oder Aktualisierung von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch. AWS CLI

Wenn Sie Tags hinzufügen, darf ein Tag-Schlüssel nicht leer sein. Der Schlüssel darf maximal 127 Zeichen lang sein und nur Unicode-Buchstaben, Ziffern oder Trennzeichen oder die folgenden Sonderzeichen enthalten: `+ - = . _ : / @`-Tag-Werte sind optional. Sie können einen Tag hinzufügen, der einen Schlüssel, aber keine Werte enthält. Der Wert darf maximal 255 Zeichen lang sein und können nur Unicode-Buchstaben, Ziffern oder Trennzeichen oder die folgenden Sonderzeichen enthalten: `+ - = . _ : / @`.

Themen

- Hinzufügen von Tags zu einem neuen AWS OpsWorks for Chef Automate Server (CLI)
- Hinzufügen von Tags zu einer neuen Sicherung (CLI)
- Hinzufügen von Tags zu vorhandenen Servern oder Sicherungen (CLI)
- Auflisten der Ressourcen-Tags
- Löschen von Tags von einer Ressource

Hinzufügen von Tags zu einem neuen AWS OpsWorks for Chef Automate Server (CLI)

Sie können das verwenden AWS CLI , um Tags hinzuzufügen, wenn Sie einen AWS OpsWorks for Chef Automate Server erstellen. In diesem Verfahren wird nicht vollständig beschrieben, wie ein Server erstellt wird. Ausführliche Informationen zum Erstellen eines AWS OpsWorks for Chef Automate Servers AWS CLI finden Sie unter Erstellen Sie einen Chef Automate-Server mit dem AWS CLI in diesem Handbuch. Sie können einem Server bis zu 50 Tags hinzufügen.

1. Stellen Sie sicher, dass alle Voraussetzungen für die Erstellung eines AWS OpsWorks for Chef Automate Servers erfüllt sind.
2. Führen Sie die Schritte 1-5 von Erstellen Sie einen Chef Automate-Server mit dem AWS CLI durch.
3. Fügen Sie beim Ausführen des `create-server`-Befehls in Schritt 6 den `--tags`-Parameter zu dem Befehl hinzu, wie im folgenden Beispiel gezeigt.

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

   Im Folgenden finden Sie ein Beispiel, das nur den Tag-Teil des `create-server`-Befehls zeigt.

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

4. Führen Sie die verbleibenden Schritte unter Erstellen Sie einen Chef Automate-Server mit dem AWS CLI aus. Führen Sie die Schritte unter Auflisten der Ressourcen-Tags in diesem Thema aus, um zu überprüfen, ob Ihre Tags dem neuen Server hinzugefügt wurden.

Hinzufügen von Tags zu einer neuen Sicherung (CLI)

Sie können den verwenden AWS CLI , um Tags hinzuzufügen, wenn Sie ein neues, manuelles Backup eines AWS OpsWorks for Chef Automate Servers erstellen. In diesem Verfahren wird nicht vollständig beschrieben, wie eine manuelle Sicherung erstellt wird. Ausführliche Informationen zum Erstellen einer manuellen Sicherung finden Sie unter „So führen Sie eine manuelle Sicherung durch AWS CLI“ im "inEinen AWS OpsWorks for Chef Automate Server sichern. Sie können einer Sicherung bis zu 50 Tags hinzufügen. Wenn ein Server über Tags verfügt, werden neue Sicherungen automatisch mit den Tags des Servers markiert.

Wenn Sie einen neuen AWS OpsWorks for Chef Automate Server erstellen, sind automatische Backups standardmäßig aktiviert. Sie können Tags zu einer automatisierten Sicherung hinzufügen, indem Sie den unter Hinzufügen von Tags zu vorhandenen Servern oder Sicherungen (CLI) in diesem Thema beschriebenen `tag-resource`-Befehl ausführen.

- Führen Sie den folgenden Befehl aus, um einer manuellen Sicherung während der Erstellung der Sicherung Tags hinzuzufügen. Nur der Tag-Teil des Befehls wird angezeigt. Ein Beispiel für den vollständigen `create-backup`-Befehl finden Sie unter „So führen Sie eine manuelle Sicherung in der AWS CLI aus" in Einen AWS OpsWorks for Chef Automate Server sichern.

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

  Das folgende Beispiel zeigt nur den Tag-Teil des `create-backup`-Befehls.

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

Hinzufügen von Tags zu vorhandenen Servern oder Sicherungen (CLI)

Sie können den `tag-resource`-Befehl ausführen, um Tags zu vorhandenen AWS OpsWorks for Chef Automate -Servern oder Sicherungen hinzuzufügen (unabhängig davon, ob die Sicherungen automatisch oder manuell erstellt wurden). Geben Sie den Amazon-Ressourcennamen (ARN) einer Zielressource an, um ihr Tags hinzuzufügen.

1. So rufen Sie den ARN der Ressource ab, auf die Sie Tags anwenden möchten:

- Führen Sie für einen Server `describe-servers --server-name` *`server_name`* aus. Die Ergebnisse des Befehls zeigen den Server-ARN an.
- Führen Sie für eine Sicherung `describe-backups --backup-id` *`backup_ID`* aus. Die Ergebnisse des Befehls zeigen den ARN der Sicherung an. Sie können auch ausführen`describe-backups --server-name` *`server_name`*, um Informationen zu allen Backups für einen bestimmten AWS OpsWorks for Chef Automate Server anzuzeigen.

Das folgende Beispiel zeigt nur die `ServerArn` in den Ergebnissen eines `describe-servers --server-name opsworks-cm-test`-Befehls an. Der `ServerArn`-Wert wird einem `tag-resource`-Befehl hinzugefügt, um dem Server Tags hinzuzufügen.

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

2. Führen Sie den `tag-resource`-Befehl mit dem ARN aus, den Sie in Schritt 1 erhalten haben.

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

Im Folgenden wird ein Beispiel gezeigt.

```
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. Um zu überprüfen, ob Tags erfolgreich hinzugefügt wurden, fahren Sie mit der nächsten Prozedur, Auflisten der Ressourcen-Tags, fort.

Auflisten der Ressourcen-Tags

Sie können den `list-tags-for-resource` Befehl ausführen, um die Tags anzuzeigen, die an AWS OpsWorks for Chef Automate Server oder Backups angehängt sind. Geben Sie den ARN einer Zielressource an, um deren Tags anzuzeigen.

1. So rufen Sie den ARN der Ressource ab, für die Sie Tags auflisten möchten:

   - Führen Sie für einen Server `describe-servers --server-name` *`server_name`* aus. Die Ergebnisse des Befehls zeigen den Server-ARN an.
   - Führen Sie für eine Sicherung `describe-backups --backup-id` *`backup_ID`* aus. Die Ergebnisse des Befehls zeigen den ARN der Sicherung an. Sie können auch ausführen`describe-backups --server-name` *`server_name`*, um Informationen zu allen Backups für einen bestimmten AWS OpsWorks for Chef Automate Server anzuzeigen.

2. Führen Sie den `list-tags-for-resource`-Befehl mit dem ARN aus, den Sie in Schritt 1 erhalten haben.

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

   Im Folgenden wird ein Beispiel gezeigt.

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

   Wenn Tags auf der Ressource vorhanden sind, gibt der Befehl Ergebnisse wie die folgenden zurück.

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

Löschen von Tags von einer Ressource

Sie können den `untag-resource`-Befehl ausführen, um Tags von AWS OpsWorks for Chef Automate -Servern oder Sicherungen zu löschen. Wenn die Ressource gelöscht wird, werden auch die Tags in der Ressource gelöscht. Geben Sie den Amazon-Ressourcennamen (ARN) einer Zielressource an, um Tags von ihr zu entfernen.

1. So rufen Sie den ARN der Ressource ab, von der Sie Tags entfernen möchten:
   - Führen Sie für einen Server `describe-servers --server-name` *`server_name`* aus. Die Ergebnisse des Befehls zeigen den Server-ARN an.
   - Führen Sie für eine Sicherung `describe-backups --backup-id` *`backup_ID`* aus. Die Ergebnisse des Befehls zeigen den ARN der Sicherung an. Sie können auch ausführen`describe-backups --server-name` *`server_name`*, um Informationen zu allen Backups für einen bestimmten AWS OpsWorks for Chef Automate Server anzuzeigen.
2. Führen Sie den `untag-resource`-Befehl mit dem ARN aus, den Sie in Schritt 1 erhalten haben. Geben Sie nur die Tags an, die Sie löschen möchten.

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

   In diesem Beispiel entfernt der `untag-resource`-Befehl nur den Tag mit dem Schlüssel `Stage` und dem Wert `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. Führen Sie die Schritte unter Auflisten der Ressourcen-Tags in diesem Thema aus, um zu überprüfen, ob Tags erfolgreich gelöscht wurden.

Weitere Informationen finden Sie unter:

- Erstellen Sie einen Chef Automate-Server mit dem AWS CLI
- Einen AWS OpsWorks for Chef Automate Server sichern
- AWS-Strategien für das Tagging

- Steuern des Zugriffs auf AWS Ressourcen mithilfe von Ressourcen-Tags im AWS Identity and Access Management Benutzerhandbuch
- Verwenden von Kostenzuordnungs-Tags im AWS Billing and Cost Management - Benutzerhandbuch.
- CreateBackup in der AWS OpsWorks CM-API-Referenz
- CreateServer in der AWS OpsWorks CM-API-Referenz
- TagResource in der AWS OpsWorks CM-API-Referenz
- ListTagsForResource in der AWS OpsWorks CM-API-Referenz
- UntagResource in der AWS OpsWorks CM-API-Referenz

Einen AWS OpsWorks for Chef Automate Server sichern und wiederherstellen

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Lebensende erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

In diesem Abschnitt wird beschrieben, wie Sie einen AWS OpsWorks for Chef Automate Server sichern und wiederherstellen und wie Sie Backups löschen.

Themen

- Einen AWS OpsWorks for Chef Automate Server sichern
- Einen AWS OpsWorks for Chef Automate Server aus einem Backup wiederherstellen

Einen AWS OpsWorks for Chef Automate Server sichern

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

Sie können ein täglich oder wöchentlich wiederkehrendes AWS OpsWorks for Chef Automate Server-Backup definieren und den Service die Backups in Ihrem Namen in Amazon Simple Storage Service (Amazon S3) speichern lassen. Alternativ können Sie bei Bedarf manuelle Sicherungen durchführen.

Da Backups in Amazon S3 gespeichert werden, fallen zusätzliche Gebühren an. Sie können einen Aufbewahrungszeitraum für Backups von bis zu 30 Generationen definieren. Sie können eine Serviceanfrage stellen, um dieses Limit ändern zu lassen, indem Sie die AWS Support-Kanäle nutzen. Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Sie können den Backups eines AWS OpsWorks for Chef Automate Servers Tags hinzufügen. Wenn Sie einem AWS OpsWorks for Chef Automate -Server Tags hinzugefügt haben, werden diese Tags von automatischen Sicherungen des Servers geerbt. Weitere Informationen zum Hinzufügen und Verwalten von Tags für Sicherungen finden Sie unter Mit Tags auf AWS OpsWorks for Chef Automate Ressourcen arbeiten in diesem Handbuch.

Themen

Automatische Backups

Wenn Sie Ihren AWS OpsWorks for Chef Automate Server konfigurieren, wählen Sie entweder automatische oder manuelle Backups. AWS OpsWorks for Chef Automate startet automatische

Backups während der Stunde und an dem Tag, den Sie im Abschnitt Automatisches Backup auf der Seite Erweiterte Einstellungen konfigurieren von Setup ausgewählt haben. Wenn Ihr Server bereits online ist, können Sie die Sicherungseinstellungen ändern, indem Sie die folgenden Schritte ausführen, entweder über die Kachel des Servers auf der Startseite der Chef Automate-Server oder auf der Eigenschaftenseite des Servers.

Ändern der Einstellungen für automatische Sicherungen

1. Wählen Sie im Menü Actions (Aktionen) der Serverkachel auf der Startseite Chef servers (Chef-Server) die Option Change settings (Einstellungen ändern) aus.
2. Um automatisierte Sicherungen zu deaktivieren, wählen Sie No (Nein) für die Option Enable automated backups (Automatische Sicherungen aktivieren) aus. Speichern Sie Ihre Änderungen. Sie müssen nicht zum nächsten Schritt gehen.
3. Ändern Sie im Abschnitt Automated Backup (Automatische Sicherung) die Häufigkeit, die Startzeit oder die Generationen, die aufbewahrt werden sollen. Speichern Sie Ihre Änderungen.

Manuelle Sicherungen

Sie können ein manuelles Backup jederzeit im oder starten AWS Management Console, indem Sie den Befehl AWS CLI create-backup ausführen. Manuelle Backups sind nicht in den maximal 30 Generationen automatisierter Backups enthalten, die gespeichert werden. Es werden maximal 10 manuelle Backups gespeichert und müssen manuell aus Amazon S3 gelöscht werden.

Sie können Tags hinzufügen, wenn Sie ein neues, manuelles Backup eines AWS OpsWorks for Chef Automate Servers erstellen. Weitere Informationen zum Hinzufügen von Tags beim Erstellen einer manuellen Sicherung finden Sie unter Hinzufügen von Tags zu einer neuen Sicherung (CLI).

Um ein manuelles Backup durchzuführen in AWS Management Console

1. Wählen Sie auf der Seite Chef Automate servers (Chef Automate-Server) den Server aus, für den Sie eine Sicherung erstellen möchten.
2. Wählen Sie auf der Eigenschaftenseite im linken Navigationsbereich die Option Backups (Sicherungen).
3. Wählen Sie Create backup (Backup erstellen).
4. Die manuelle Sicherung ist fertig, wenn auf der Seite ein grünes Häkchen in der Spalte Status der Sicherung angezeigt wird.

Um ein manuelles Backup durchzuführen in AWS CLI

- Führen Sie den folgenden AWS CLI Befehl aus, um ein manuelles Backup zu starten.

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

Sicherungen löschen

Das Löschen einer Sicherung löscht diese endgültig aus dem S3-Bucket, in dem Sicherungen gespeichert werden.

Um ein Backup in der AWS Management Console

1. Wählen Sie auf der Seite Chef Automate servers (Chef Automate-Server) den Server aus, für den Sie eine Sicherung erstellen möchten.
2. Wählen Sie auf der Eigenschaftenseite im linken Navigationsbereich die Option Backups (Sicherungen).
3. Wählen Sie die Sicherung, die Sie löschen möchten, und wählen Sie dann Delete backup (Sicherung löschen). Sie können jeweils nur eine Sicherung auswählen.
4. Wenn Sie aufgefordert werden, das Löschen zu bestätigen, markieren Sie das Kontrollkästchen für Delete the backup, which is stored in an S3 bucket (Sicherung löschen, die in einem S3-Bucket gespeichert ist) und wählen Sie dann Yes, Delete (Ja, löschen).

Um ein Backup zu löschen in AWS CLI

- Um ein Backup zu löschen, führen Sie den folgenden AWS CLI Befehl aus und `--backup-id` ersetzen Sie es durch die ID des Backups, das Sie löschen möchten. Backup-IDs haben das Format *ServerName-yyyymmddhhmmssssss*. z. B. **`test-chef-server-20171218132604388`**.

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

Einen AWS OpsWorks for Chef Automate Server aus einem Backup wiederherstellen

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Lebensende erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

Nachdem du deine verfügbaren Backups durchsucht hast, kannst du einen Zeitpunkt wählen, ab dem du deinen AWS OpsWorks for Chef Automate Server wiederherstellen möchtest. Serversicherungen enthalten nur für Konfigurationsmanagementsoftware persistente Daten (Rezeptbücher, registrierte Knoten usw.). Wenn Sie eine direkte Wiederherstellung eines Servers durchführen (d. h. den vorhandenen AWS OpsWorks for Chef Automate Server auf einer neuen EC2-Instance wiederherstellen), werden Knoten, die zum Zeitpunkt des Backups registriert waren, mit dem Sie den Server wiederherstellen, erneut registriert wurden, und der Datenverkehr wird auf die neue Instanz umgeleitet, wenn die Wiederherstellung erfolgreich ist und der wiederhergestellte Serverstatus lautet. AWS OpsWorks for Chef Automate `Healthy` Bei der Wiederherstellung auf einem neu erstellten AWS OpsWorks for Chef Automate -Server werden keine Knotenverbindungen beibehalten. Beim Wiederherstellen eines Servers werden Nebenversionen der Chef-Software nicht aktualisiert. Es gelten dieselbe Chef-Version und dieselben Konfigurationsmanagement-Daten, die im Umfang der gewählten Sicherung verfügbar sind.

Die Wiederherstellung eines Servers nimmt in der Regel mehr Zeit in Anspruch als die Erstellung eines neuen Servers. Die Dauer hängt von der Größe des ausgewählten Backups ab. Nach Abschluss der Wiederherstellung bleibt die alte EC2-Instance im Zustand `Running` oder `Stopped`, jedoch nur vorübergehend. Dieser Zustand wird letztendlich beendet.

In dieser Version können Sie den verwenden, AWS CLI um einen Chef-Server in wiederherzustellen AWS OpsWorks for Chef Automate.

> **Note**
>
> Sie können auch den Befehl restore-server verwenden, um den aktuellen Instance-Typ zu ändern oder Ihren SSH-Schlüssel wiederherzustellen oder festzulegen, wenn er verloren ging oder beschädigt wurde.

Wiederherstellen eines Servers von einer Sicherung

1. Führen Sie in der den folgenden Befehl aus AWS CLI, um eine Liste der verfügbaren Backups und ihrer IDs zurückzugeben. Notieren Sie sich die ID der Sicherung, die Sie verwenden möchten. Backup-IDs haben das Format *myServerName-yyyymmddhhmmssss*.

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

2. Führen Sie den folgenden Befehl aus.

   ```
   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"
   ```

   Im Folgenden wird ein Beispiel gezeigt.

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

3. Warten Sie, bis die Wiederherstellung abgeschlossen ist.

Systemwartung in AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

Durch die obligatorische Systemwartung wird sichergestellt, dass die neuesten Nebenversionen von Chef Server und Chef Automate Server, einschließlich Sicherheitsupdates, immer auf einem AWS OpsWorks for Chef Automate Server laufen. Die Systemwartung muss mindestens einmal pro Woche durchgeführt werden. Mit dem AWS CLI können Sie auf Wunsch die tägliche automatische Wartung konfigurieren. Sie können den auch verwenden AWS CLI , um neben der planmäßigen Systemwartung auch Systemwartungen bei Bedarf durchzuführen.

Wenn neue Nebenversionen der Chef-Software verfügbar werden, aktualisiert die Systemwartung die Nebenversion von Chef Automate und Chef Server automatisch auf dem Server, wenn sie den AWS-Test bestanden hat. AWS führt umfangreiche Tests durch, um sicherzustellen, dass Chef-Upgrades produktionsbereit sind und bestehende Kundenumgebungen nicht stören. Daher kann es zu Verzögerungen zwischen den Chef-Softwareversionen und ihrer Verfügbarkeit für die Anwendung auf bestehenden OpsWorks Chef Automate-Servern kommen. Weitere Informationen zum Aktualisieren verfügbarer Nebenversionen von Chef-Software auf Anfrage finden Sie Starten der Systemwartung nach Bedarf in diesem Thema.

Bei der Systemwartung wird eine neue Instance aus einem Backup gestartet, das im Rahmen des Wartungsprozesses durchgeführt wird. Dadurch wird das Risiko verringert, dass Amazon EC2 EC2-Instances, die regelmäßig gewartet werden, herabgesetzt oder beeinträchtigt werden.

> **Important**
>
> Die Systemwartung löscht alle Dateien oder benutzerdefinierten Konfigurationen, die Sie dem AWS OpsWorks for Chef Automate -Server hinzugefügt haben. Weitere Informationen zum Reparieren von Konfigurationen oder Wiederherstellen von Dateien finden Sie unter Wiederherstellung benutzerdefinierter Konfigurationen und Dateien nach der Wartung in diesem Thema.

Themen

Sicherstellen, dass die Knoten der Zertifizierungsstelle AWS OpsWorks vertrauen

> **Note**
>
> Die Schritte in diesem Abschnitt sind nicht erforderlich, wenn Sie eine benutzerdefinierte Domäne und ein benutzerdefiniertes Zertifikat mit Ihrem AWS OpsWorks for Chef Automate Server verwenden.

Knoten, die Sie mit einem AWS OpsWorks for Chef Automate Server verwalten, müssen sich mithilfe von Zertifikaten beim Server authentifizieren. AWS OpsWorks Ersetzt während der Systemwartung die Serverinstanz und generiert neue Zertifikate über die AWS OpsWorks Zertifizierungsstelle (CA) neu. Um die Kommunikation mit Ihren verwalteten Knoten nach Abschluss der Wartung automatisch wiederherzustellen, sollten die Knoten der AWS OpsWorks Zertifizierungsstelle vertrauen, die im Starterkit enthalten ist und in den Regionen gehostet wird, die von AWS OpsWorks for Chef Automate unterstützt werden. Wenn Sie die AWS OpsWorks Zertifizierungsstelle verwenden, um die Vertrauensstellung zwischen Knoten und Server herzustellen, stellen die Knoten nach der Wartung wieder eine Verbindung zur neuen Serverinstanz her. Wenn Sie EC2-Knoten mithilfe des unter beschriebenen `userdata` EC2-Skripts hinzufügen, sind die Knoten bereits so konfiguriertAutomatisches Hinzufügen von Knoten AWS OpsWorks for Chef Automate, dass sie der CA vertrauen. AWS OpsWorks

- Für Linux-basierte Knoten ist der S3-Bucket-Speicherort der Zertifizierungsstelle `https://opsworks-cm-${REGION}-prod-default-assets.s3.amazonaws.com/misc/opsworks-cm-ca-2020-root.pem`. Die AWS OpsWorks vertrauenswürdige Zertifizierungsstelle muss im Pfad gespeichert werden. `/etc/chef/opsworks-cm-ca-2020-root.pem`
- Für Windows-basierte Knoten ist der S3-Bucket-Speicherort der Zertifizierungsstelle `https://opsworks-cm-$env:AWS_REGION-prod-default-assets.s3.amazonaws.com/misc/opsworks-cm-ca-2020-root.pem`. Die AWS OpsWorks CA muss im Chef-Stammordner gespeichert sein; zum Beispiel `C:\chef\opsworks-cm-ca-2020-root.pem`

Bei beiden Pfaden wird die Regionsvariable wie folgt aufgelöst.

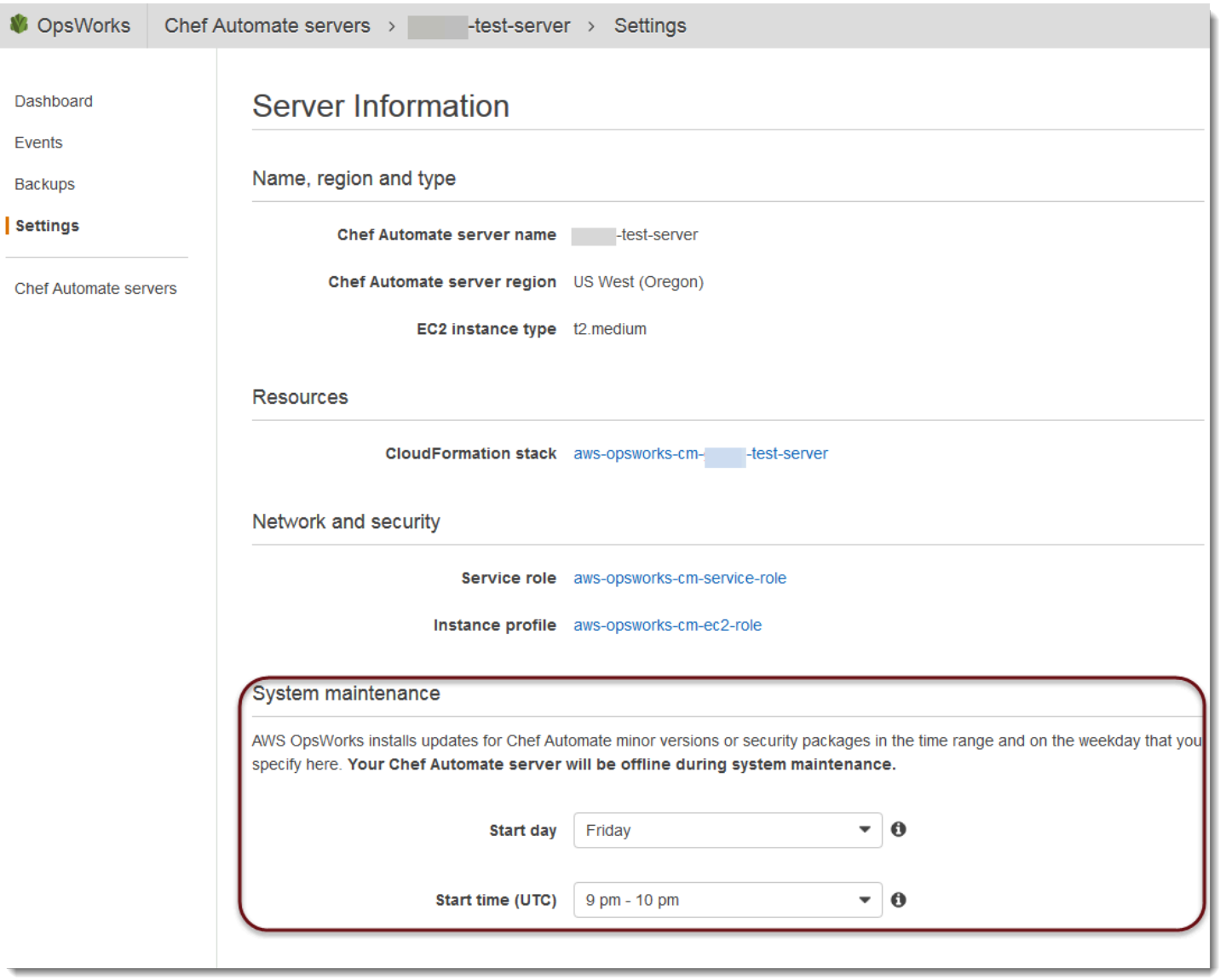
- `us-east-2`
- `us-east-1`

- us-west-1
- us-west-2
- ap-northeast-1
- ap-southeast-1
- ap-southeast-2
- eu-central-1
- eu-west-1

Konfigurieren der Systemwartung

Wenn Sie einen neuen AWS OpsWorks for Chef Automate Server erstellen, können Sie einen Wochentag und eine Uhrzeit in koordinierter Weltzeit (Coordinated Universal Time, UTC) für den Beginn der Systemwartung konfigurieren. Die Wartung beginnt während der Stunde, die Sie angeben. Da der Server während der Systemwartung offline ist, wählen Sie eine Uhrzeit innerhalb der normalen Geschäftszeiten mit geringer Server-Nachfrage aus. Der Serverstatus ist `UNDER_MAINTENANCE`, während die Wartung läuft.

Sie können auch die Systemwartungseinstellungen auf einem vorhandenen AWS OpsWorks for Chef Automate Server ändern, indem Sie die Einstellungen im Bereich Systemwartung der Einstellungsseite für Ihren Server ändern, wie im folgenden Screenshot gezeigt.



Legen Sie im Abschnitt System maintenance (Systemwartung) den Tag und die Uhrzeit fest, zu der die Systemwartung beginnen soll.

Konfiguration der Systemwartung mit dem AWS CLI

Sie können die automatische Startzeit der Systemwartung auch mithilfe der AWS CLI konfigurieren. AWS CLI Damit können Sie bei Bedarf die tägliche automatische Wartung konfigurieren, indem Sie das dreistellige Wochentagspräfix weglassen.

Fügen Sie in einem `create-server`-Befehl den Parameter `--preferred-maintenance-window` Ihrem Befehl hinzu, nachdem Sie die Anforderungen zum Erstellen der Server-Instance angegeben haben (z. B. Instance-Typ, Instance-Profil-ARN und Service-Rollen-ARN). Im folgenden `create-server`-Beispiel ist `--preferred-maintenance-window` auf `Mon:08:00` eingestellt.

Das bedeutet, dass Sie den Start der Wartung für jeden Montag um 08:00 Uhr festgelegt haben. festgelegt.

```
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 einem `update-server`-Befehl können Sie ggf. allein den Wert `--preferred-maintenance-window` aktualisieren. Im folgenden Beispiel wird das Wartungsfenster auf Freitag um 18:15 Uhr festgelegt. festgelegt.

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

Um den Beginn des Wartungsfensters auf jeden Tag um 18:15 Uhr (UTC) zu ändern, lassen Sie das aus drei Zeichen bestehende Präfix für den Wochentag weg, wie im folgenden Beispiel gezeigt.

```
aws opsworks-cm update-server --server-name "shiny-kitchen" --preferred-maintenance-
window "18:15"
```

Weitere Informationen zum Einstellen des bevorzugten Systemwartungsfensters mithilfe von finden Sie unter create-server AWS CLI und update-server.

Starten der Systemwartung nach Bedarf

Um die Systemwartung bei Bedarf außerhalb der konfigurierten wöchentlichen oder täglichen automatischen Wartung zu starten, führen Sie den folgenden Befehl aus. AWS CLI Sie können die Wartung nach Bedarf nicht in der AWS Management Console starten.

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

Weitere Informationen über diesen Befehl finden Sie unter start-maintenance.

Wiederherstellung benutzerdefinierter Konfigurationen und Dateien nach der Wartung

Bei der Systemwartung können benutzerdefinierte Dateien oder Konfigurationen, die Sie Ihrem AWS OpsWorks for Chef Automate Server hinzugefügt haben, gelöscht oder geändert werden.

Wenn nach einem Wartungslauf auf Ihrem Chef-Server Dateien oder Einstellungen fehlen, die Sie mithilfe von `RunCommand` oder SSH hinzugefügt haben, können Sie ein Amazon Machine Image (AMI) verwenden, um eine neue Amazon EC2 EC2-Instance zu starten. Es stehen AMIs zur Verfügung, die aus der Konfiguration eines Servers vor der Wartung erstellt wurden.

Die neue Instance befindet sich in demselben Zustand, in dem sich der Chef-Server vor der Wartung befand, und sollte Ihre fehlenden Dateien und Einstellungen enthalten.

> **Important**
>
> Sie können die neue Instance nicht verwenden, um Ihren Server wiederherzustellen; die Instance kann nicht als Chef-Server ausgeführt werden. Sie können die Instance nur verwenden, um Ihre Dateien und Konfigurationseinstellungen wiederherzustellen.

Um eine EC2-Instance von einem AMI aus zu starten, öffnen Sie in der Amazon EC2 EC2-Konsole den Startassistenten, wählen Sie Meine AMIs und dann das AMI aus, das Ihren Servernamen hat. Folgen Sie den Schritten des Amazon EC2 EC2-Assistenten wie bei jedem anderen Instance-Start.

Konformitätsscans in AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Lebensende erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

Mit Compliance-Scans können Sie die Compliance verwalteter Knoten in Ihrer Infrastruktur basierend auf vordefinierten Richtlinien nachverfolgen, die auch als Regeln bezeichnet werden. Compliance-

Ansichten ermöglichen Ihnen, Ihre Anwendungen regelmäßig auf Schwachstellen und nicht konforme Konfigurationen zu überprüfen. Chef bietet über 100 vordefinierter Compliance-Profile – Regelsammlungen, die für bestimmte Knotenkonfigurationen gelten –, die Sie für Ihre Compliance-Scans verwenden können. Sie können auch die InSpec Chef-Sprache verwenden, um Ihre eigenen benutzerdefinierten Profile zu erstellen.

Wenn auf Ihrem Server noch nicht Chef Automate 2.0 ausgeführt wird, können SieChef-Compliancemanuell einrichten, indem Sie das Audit-Rezeptbuch installieren.

> **Note**
>
> Die unterstützte Mindestversion der Chef Infra-Client-Agent-Software (`chef-client`) auf Knoten, die einem AWS OpsWorks for Chef Automate Server zugeordnet sind, ist 13. x. Wir empfehlen, die aktuellste, stabilste `chef-client` Version oder mindestens 14.10.9 zu verwenden.

Themen

Compliance in Chef Automate 2.0

Wenn auf Ihrem AWS OpsWorks for Chef Automate Server Chef Automate 2.0 ausgeführt wird, richten Sie Chef Compliance mithilfe der Verfahren in diesem Abschnitt ein.

Ausführen von Compliance Scan-Aufgaben mit Chef Automate 2.0

Chef Automate 2.0 beinhaltet die Funktion Chef InSpec Compliance-Scanning, für die früher eine manuelle Einrichtung und Kochbuchkonfiguration erforderlich war. Sie können Scanjobs auf einem AWS OpsWorks for Chef Automate Server ausführen, auf dem Chef Automate 2.0 ausgeführt wird. Aufgaben können (einmalig) sofort ausgeführt, für einen späteren Zeitpunkt geplant oder in regelmäßigen Abständen, beispielsweise täglich oder alle zwei Stunden, ausgeführt werden. Die Ergebnisse einer Scan-Aufgabe werden an die Compliance-Berichterstellung gesendet. Sie können

die Ergebnisse des Compliance-Scans im Dashboard von Chef Automate anzeigen und Maßnahmen dazu ergreifen. Wenn Sie die Registerkarte Compliance öffnen und Berichte auf der Registerkarte Scan-Aufgaben im Chef Automate-Dashboard anzeigen möchten, wählen rechts neben derSpalte verwalteter Knoten die Option Bericht aus.

Für Scan-Aufgaben auf verwalteten Knoten ist Folgendes erforderlich:

- Mindestens ein in Ihrem Namespace installiertes Compliance-Profil.
- Mindestens einen manuell hinzugefügten Zielknoten oder eine EC2-Instance die automatisch hinzugefügt wurde.

AWS OpsWorks for Chef Automate In werden Scanjobs auf den folgenden Zielen unterstützt.

- Manuell hinzugefügte Knoten
- `aws-ec2`-Instances
- AWS-Regionen

Detaillierte Anweisungen zum Ausführen von Scan-Aufgaben finden Sie unter Scan-Aufgaben in Chef Automate in der Chef-Dokumentation.

(Optional, Chef Automate 2.0) Einrichten von Compliance mit dem Audit-Rezeptbuch

Sie können die Konformität auf jedem AWS OpsWorks for Chef Automate Server konfigurieren. Nachdem Sie einen AWS OpsWorks for Chef Automate -Server gestartet haben, können Sie Profile aus dem Chef Automate-Dashboard installieren, oder die gewünschten Profile den Audit-Rezeptbuchattributen in der `Policyfile.rb`-Richtliniendatei hinzufügen. Eine vorbelegte `Policyfile.rb`-Datei ist im Starter Kit enthalten.

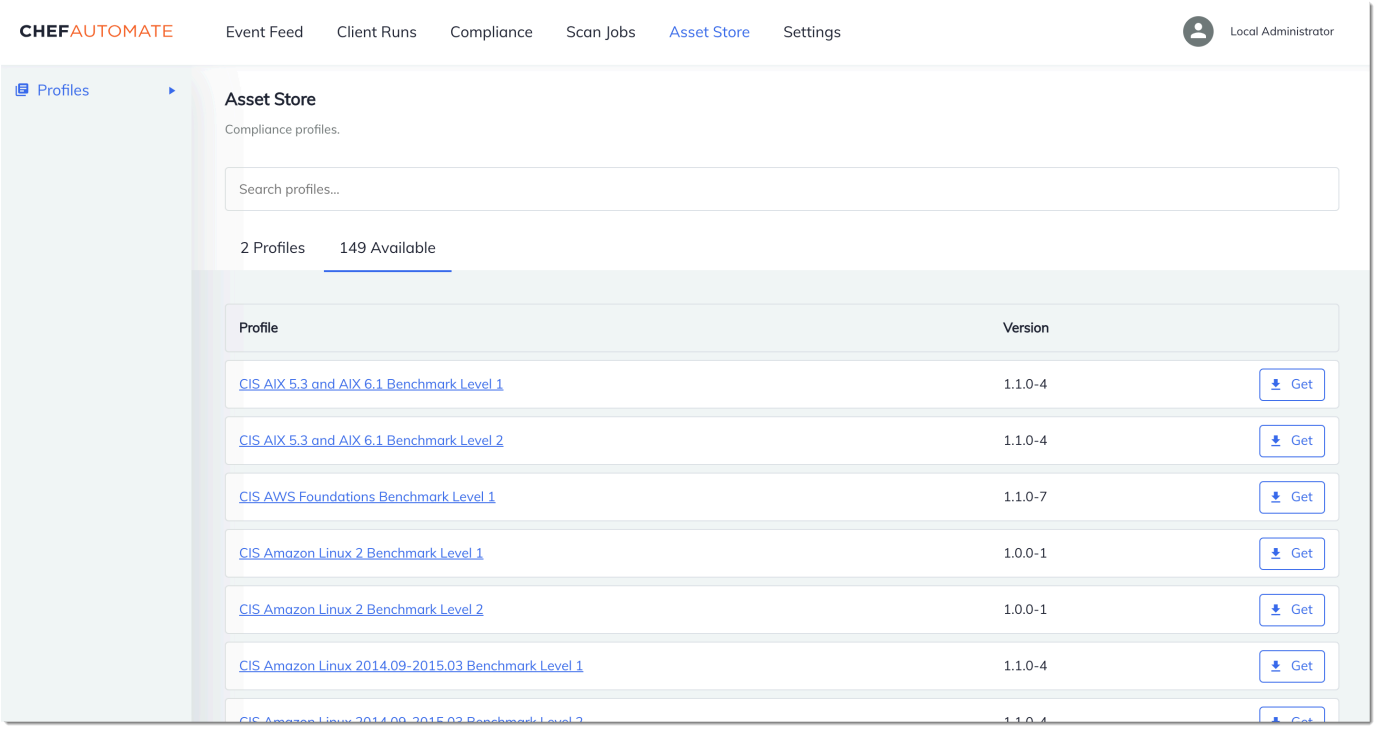
Nach dem Sie `Policyfile.rb` mit Profilen als Attribute des Audit-Rezeptbuchs bearbeitet haben, führen Sie `chef push`-Befehle zum Hochladen des Audit-Rezeptbuchs und anderer in `Policyfile.rb` angegebener Rezeptbücher auf Ihren Chef Automate-Server hoch. Durch die Installation des Audit-Kochbuchs wird auch das Gem für Chef installiert InSpec, ein Open-Source-Framework für Tests und Prüfungen, das von Chef entwickelt wurde. Für Chef Automate 2.0 wählen Sie Version 7.1.0 des Audit-Rezeptbuchs oder höher aus. Das InSpec Gem muss Version 2.2.102 oder höher sein.

Die Anweisungen in diesem Abschnitt verdeutlichen, wie Sie das `opsworks-audit`-Rezeptbuch implementieren. Das Audit-Kochbuch lädt bestimmte Profile vom Chef Automate-Server herunter,

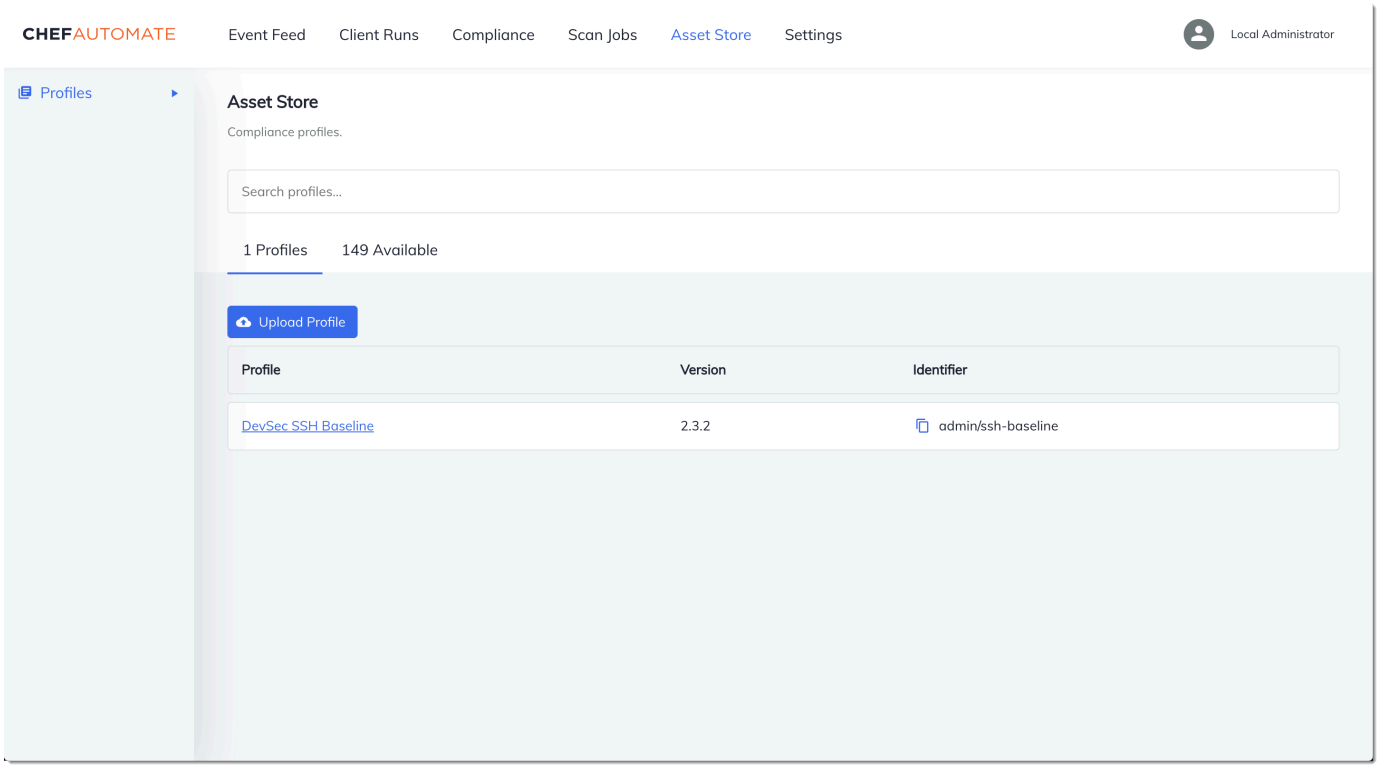
bewertet Knoten anhand des DevSec SSH-Baseline-Profils und meldet bei jedem Lauf das Ergebnis der Konformitätsscans. `chef-client`

So installieren Sie Compliance-Profile

1. Melden Sie sich beim web-basierten Dashboard von Chef Automate an, falls Sie dies noch nicht getan haben. Verwenden Sie die Anmeldeinformationen, die Sie beim Herunterladen des Starter Kits während der Erstellung Ihres AWS OpsWorks for Chef Automate -Servers erhalten haben.
2. Wählen Sie im Chef Automate-Dashboard die Registerkarte Asset Store (Objekt-Store) aus.



3. Wählen Sie die Registerkarte Available (Verfügbar) aus, um vordefinierte Profile anzuzeigen.
4. Durchsuchen Sie die Liste der Profile. Wählen Sie ein Profil, das mit dem Betriebssystem und der Konfiguration von mindestens einem Ihrer verwalteten Knoten übereinstimmt. Um Details über das Profil anzuzeigen, einschließlich einer Beschreibung der Verletzungen, auf die das Profil ausgelegt ist, und des zugrunde liegenden Regelcodes, wählen Sie > rechts neben dem Profileintrag. Sie können mehrere Profile auswählen. Wenn Sie das Beispiel im Starter Kit einrichten, wählen Sie DevSec SSH Baseline.



5. Um die ausgewählten Profile auf Ihrem Chef Automate-Server zu installieren, wählen Sie Get (Abrufen).
6. Nachdem Sie Profile installiert haben, werden sie in der Registerkarte Profiles (Profile) des Chef Automate-Dashboards angezeigt.

So installieren Sie Rezeptbücher mit **`Policyfile.rb`**

1. Zeigen Sie `Policyfile.rb` in Ihrem Starter Kit an, um zu sehen, dass die Attribute für das Audit-Rezeptbuch das `ssh-baseline`-Profil in `['profiles']` angeben.

```
# 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. Laden Sie die in `Policyfile.rb` definierten Rezeptbücher herunter und installieren Sie sie.

```
chef install
```

Alle Rezeptbücher sind in der `metadata.rb`-Datei des Rezeptbuchs versioniert. Immer wenn Sie ein Rezeptbuch ändern, müssen Sie die Version des Rezeptbuchs ändern, die sich in seiner `metadata.rb` befindet.

3. Leiten Sie die in `Policyfile.rb` definierte `opsworks-demo`-Richtlinie an Ihren Server weiter.

```
chef push opsworks-demo
```

4. Überprüfen Sie die Installation Ihrer Richtlinie. Führen Sie den folgenden Befehl aus.

```
chef show-policy
```

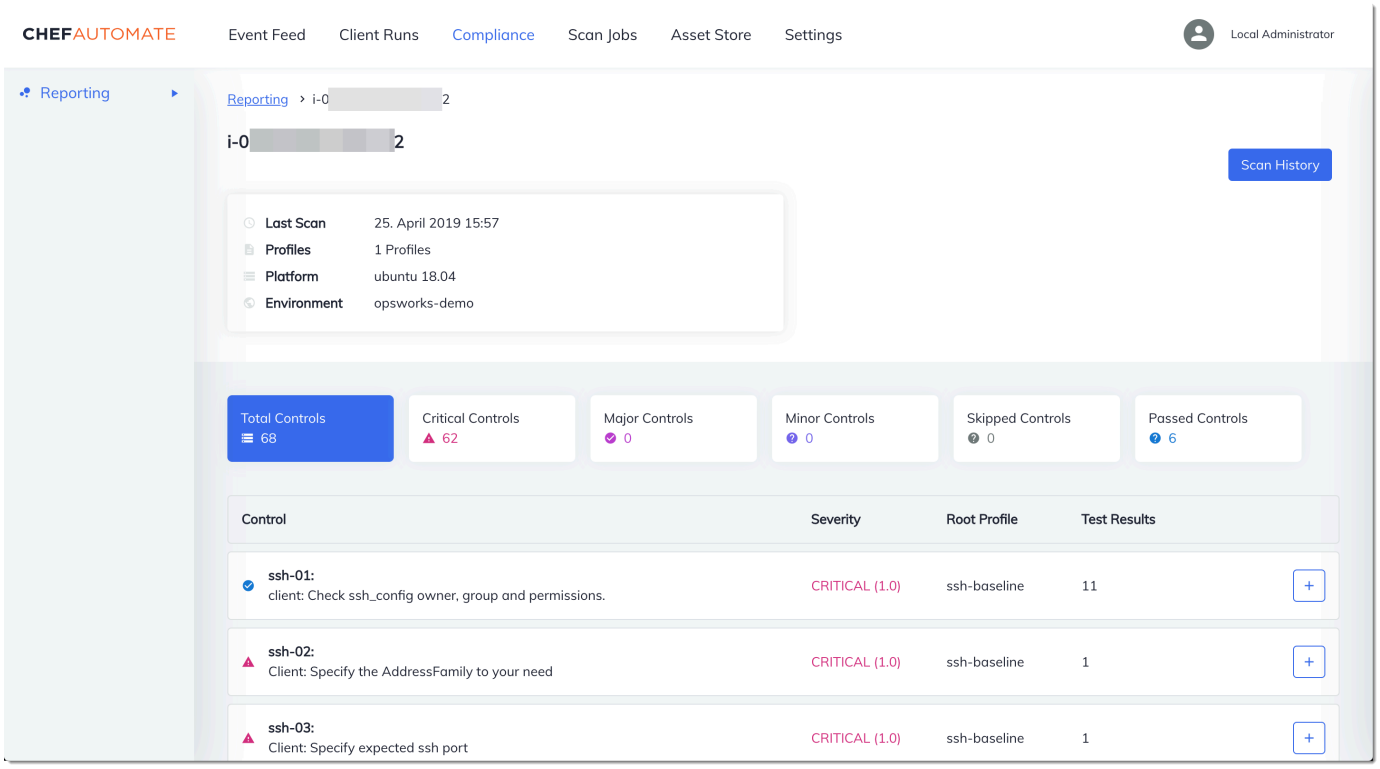
Die Ergebnisse sollten etwa wie folgt aussehen:

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

5. Fügen Sie Ihrem Server zu verwaltende Knoten hinzu, sofern Sie dies nicht bereits getan haben. Verwenden Sie das `userdata.sh` Skript, das in diesem Starterkit enthalten ist, um Ihren ersten Knoten mit dem AWS OpsWorks for Chef Automate Server zu verbinden. Es verwendet die AWS OpsWorks `AssociateNode` API, um einen Knoten mit Ihrem Server zu verbinden.

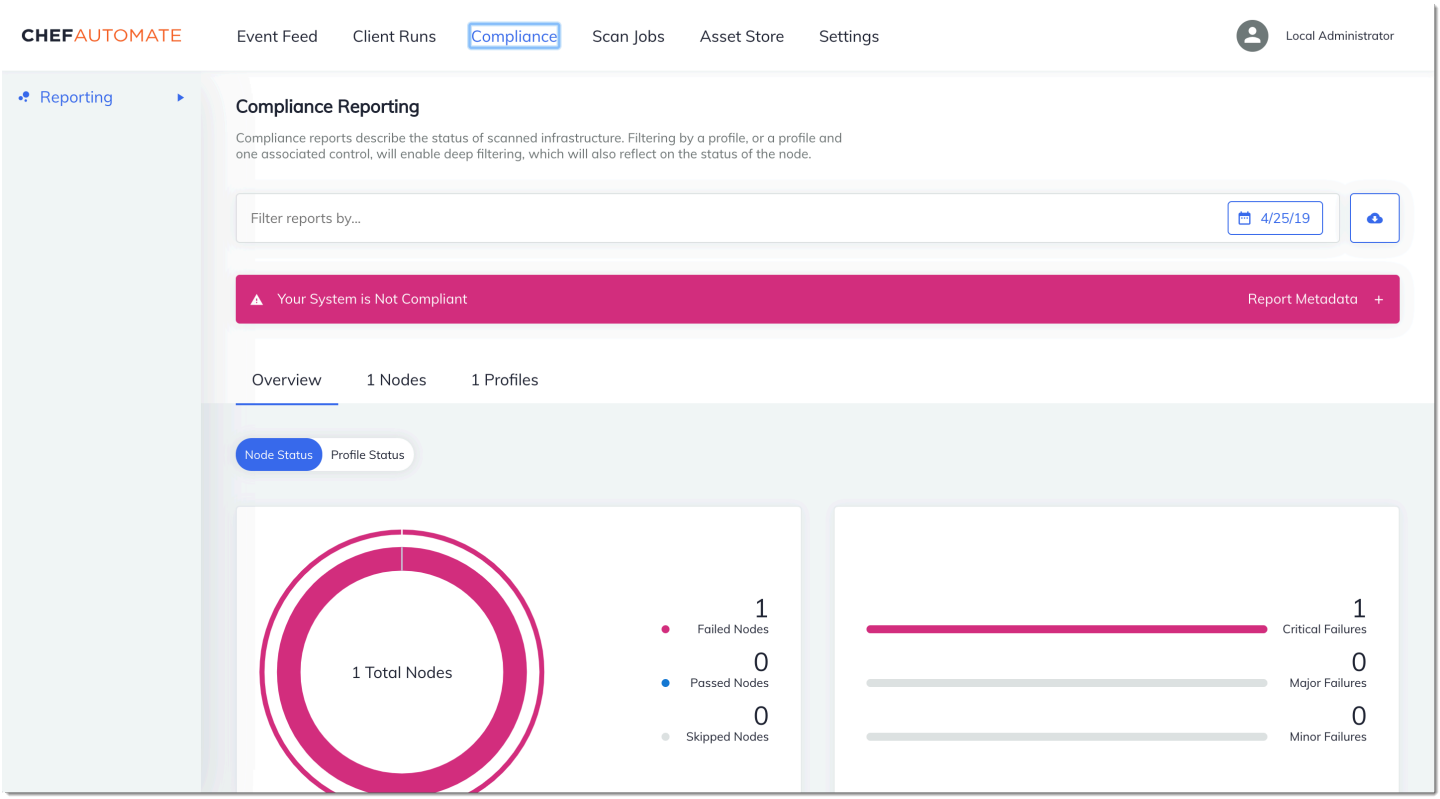
   Sie können die Zuordnung von Knoten automatisieren, indem Sie die Schritte in Automatisches Hinzufügen von Knoten AWS OpsWorks for Chef Automate ausführen, oder Knoten einzeln hinzufügen, indem Sie die Schritte in Fügen Sie Knoten einzeln hinzu ausführen.

6. Nachdem Sie die Ausführungsliste für Ihre Knoten aktualisiert haben, führt der `chef-client`-Agent bei der nächsten Ausführung die von Ihnen spezifizierten Rezepte aus. Dies erfolgt standardmäßig alle 1800 Sekunden (30 Minuten). Nach der Ausführung können Sie Compliance-Ergebnisse in der Registerkarte Compliance auf dem Chef Automate-Dashboard anzeigen und entsprechende Maßnahmen ergreifen.

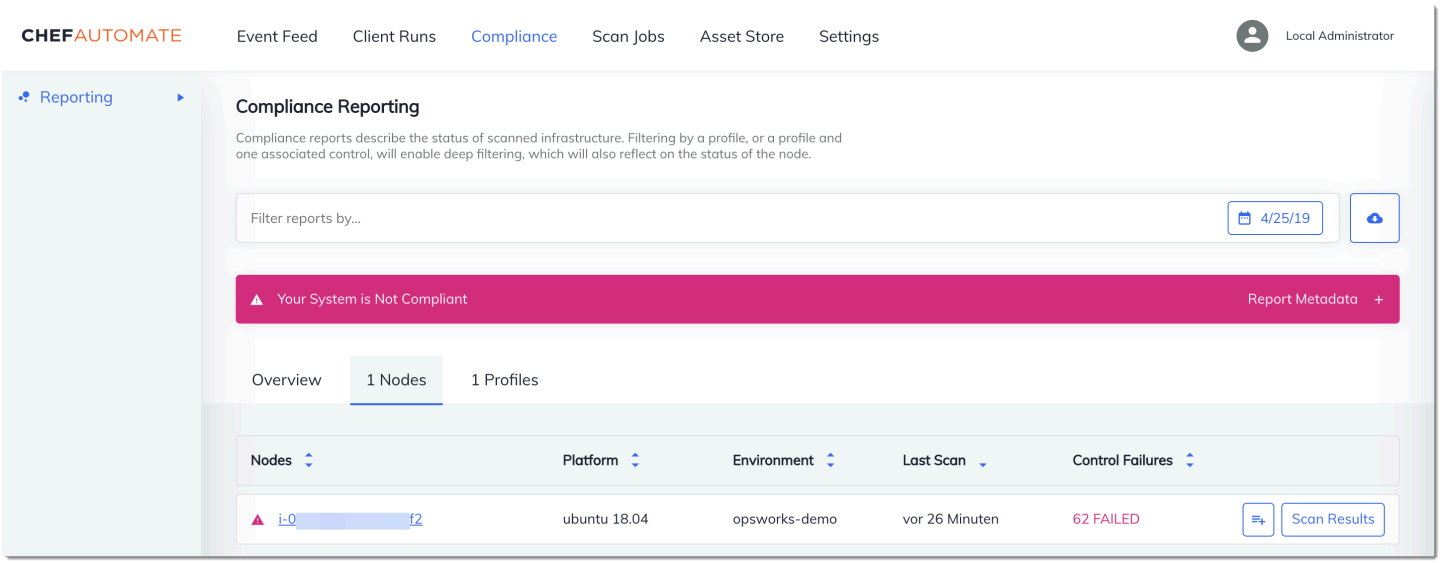


Ausführen eines Compliance-Scans

Sie sollten die Ergebnisse des Compliance-Scans kurz nach der ersten Ausführung des Agenten-Daemons nach der Konfiguration Knoten-Ausführungslisten im Dashboard von Chef Automate sehen.



Wählen Sie im Chef Automate-Dashboard die Registerkarte Compliance. Wählen Sie im linken Navigationsbereich die Option Reporting (Berichterstellung) aus. Wählen Sie auf die Registerkarte Profiles (Profile), dort Scan Results (Scan-Ergebnisse) und dann einen Knoten mit Scan-Fehlern aus, um mehr über die Regeln zu erfahren, gegen die ein Knoten verstoßen hat.



In der Regel werden fehlerhafte Scanergebnisse angezeigt, da neue Knoten noch nicht alle Regeln im DevSec SSH-Baseline-Profil erfüllen. Das DevSec Hardening Framework, ein Community-Projekt,

bietet Kochbücher zur Behebung von Problemen, die gegen die Regeln im SSH-Baseline-Profil verstoßen. DevSec

(Optional) Auflösen nicht konformer Ergebnisse

Das Starterkit enthält ein Open-Source-Kochbuch, das Sie ausführen können**`ssh-hardening`**, um fehlerhafte Ergebnisse von Läufen gegen das SSH-Baseline-Profil zu korrigieren. DevSec

> **Note**
>
> Das **`ssh-hardening`** Cookbook nimmt Änderungen an Ihren Knoten vor, um den SSH-Baseline-Regeln zu entsprechen. DevSec Bevor Sie dieses Kochbuch auf Produktionsknoten ausführen, überprüfen Sie die Details zum DevSec SSH-Baseline-Profil in der Chef Automate-Konsole, um zu verstehen, auf welche Regelverstöße das Cookbook abzielt. Sehen Sie sich die Informationen über das als Open-Source bereitgestellte `ssh-hardening`-Rezeptbuch an, bevor Sie es auf Produktionsknoten ausführen.

So führen Sie das **`ssh-hardening`**-Rezeptbuch aus

1. Wesen Sie in einem Texteditor das `ssh-hardening`-Rezeptbuch einer `Policyfile.rb`-Ausführungsliste zu. Die `Policyfile.rb`-Ausführungsliste sollte wie folgt aussehen.

   ```
   run_list  'chef-client', 'opsworks-webserver', 'audit', 'ssh-hardening'
   ```

2. Aktualisieren Sie `Policyfile.rb` und leiten Sie die Datei an Ihrem AWS OpsWorks for Chef Automate -Server weiter.

   ```
   chef update Policyfile.rb
      chef push opsworks-demo
   ```

3. Knoten, der die `opsworks-demo`-Richtlinien zugewiesen sind, aktualisieren die Ausführungsliste automatisch und wenden das `ssh-hardening`-Rezeptbuch bei der nächsten Ausführung von `chef-client` an.

   Da Sie das `chef-client`-Rezeptbuch verwenden, meldet sich Ihr Knoten in regelmäßigen Zeitabständen an (standardmäßig alle 30 Minuten). Beim nächsten Check-in wird das `ssh-hardening` Cookbook ausgeführt und trägt dazu bei, die Knotensicherheit zu verbessern, um die Regeln des DevSec SSH-Baseline-Profils zu erfüllen.

4. Warten Sie nach der ersten Ausführung des `ssh-hardening`-Rezeptbuchs 30 Minuten, bevor Sie wieder einen Compliance-Scan ausführen. Sehen Sie sich die Ergebnisse im Chef Automate-Dashboard an. Die fehlerhaften Ergebnisse, die beim ersten Durchlauf des DevSec SSH-Baseline-Scans aufgetreten sind, sollten behoben werden.

Compliance in Chef Automate 1.x

Wenn auf Ihrem AWS OpsWorks for Chef Automate Server Chef Automate 1 ausgeführt wird. x, richten Sie Chef Compliance mithilfe der Verfahren in diesem Abschnitt ein.

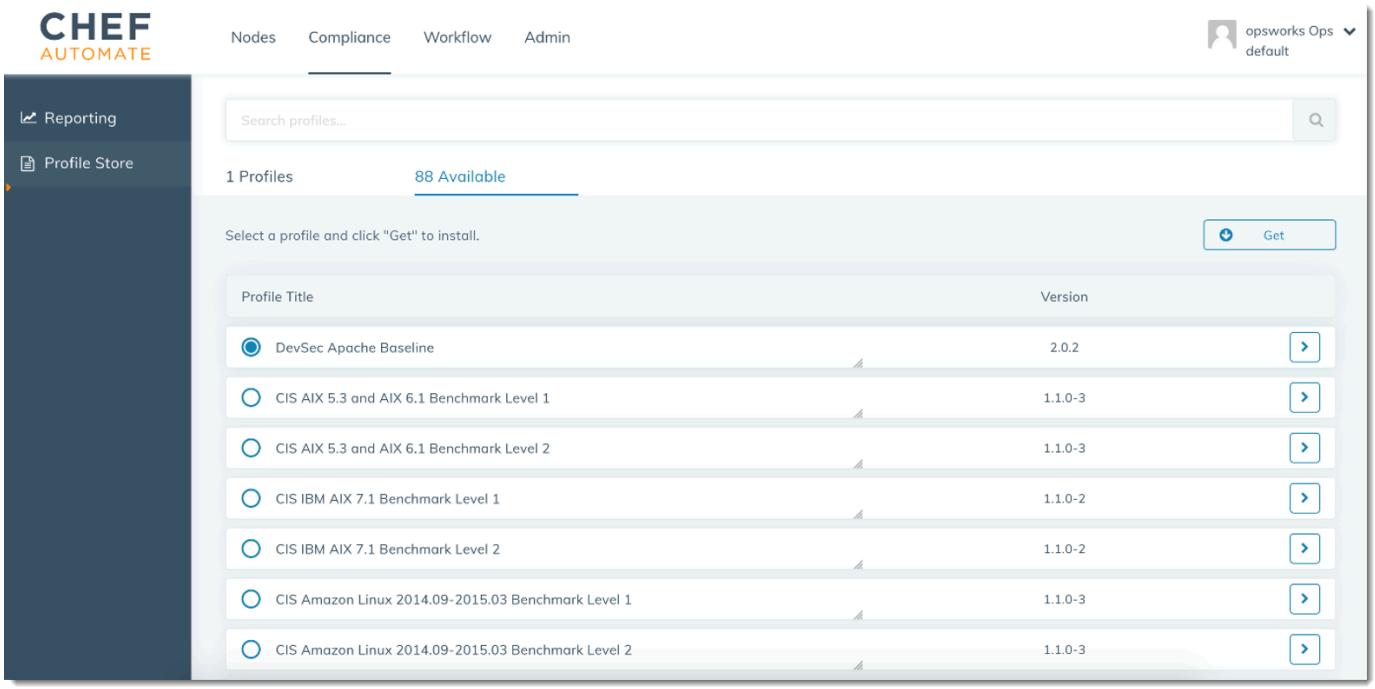
(Optional, Chef Automate 1.x) Einrichten von Chef-Compliance

Sie können Chef Compliance auf jedem AWS OpsWorks for Chef Automate Server konfigurieren. Nachdem Sie einen AWS OpsWorks for Chef Automate -Server gestartet haben, wählen Sie Profile aus, die Sie von den Profilen auf dem Chef Automate-Dashboard aus ausführen wollen. Nachdem Sie Profile installiert haben, führen Sie `berks`-Befehle zum Hochladen des Audit-Rezeptbuchs auf Ihren Chef Automate-Server aus. Durch die Installation des Audit-Kochbuchs wird auch das Gem for installiert InSpec, ein von Chef entwickeltes Open-Source-Test-Framework, mit dem Sie automatisierte Tests in jede Phase Ihrer Bereitstellungspipeline integrieren können. Für Chef Automate 1.x wählen Sie Version 5.0.1 oder höher des Audit-Rezeptbuchs aus. Das InSpec Gem muss Version 1.24.0 oder höher sein.

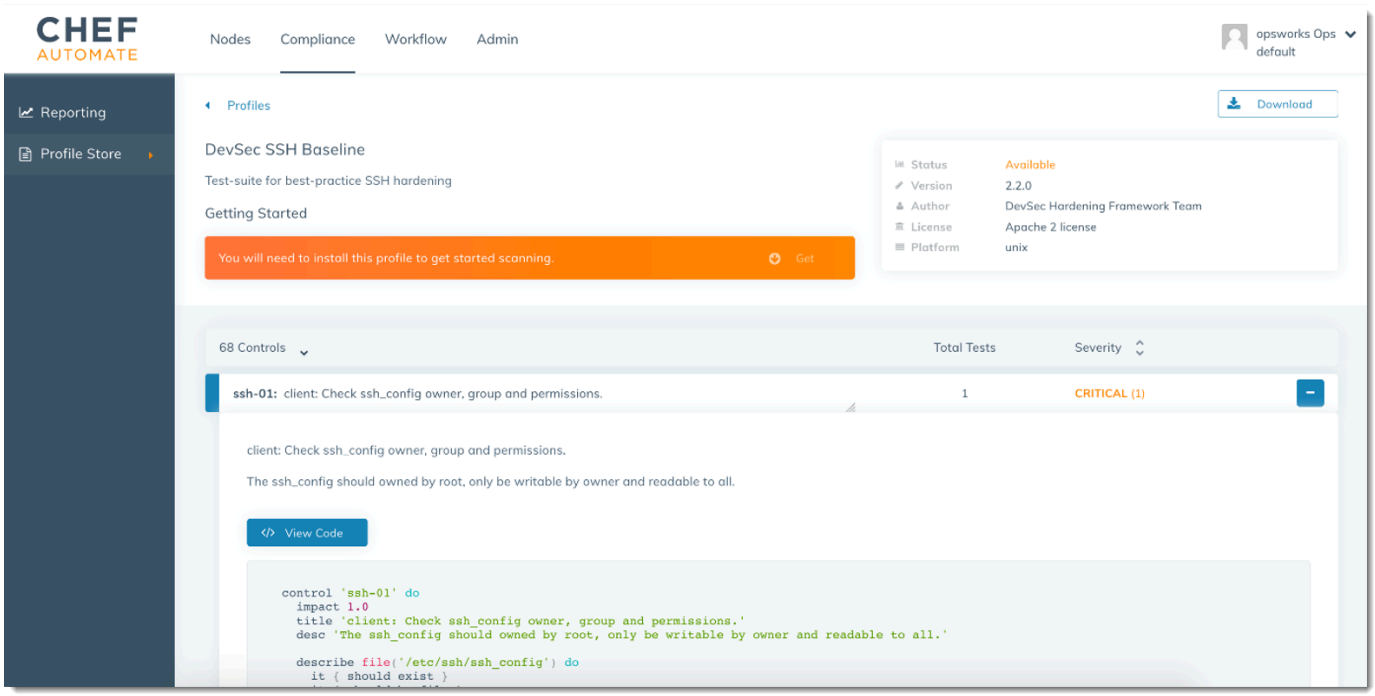
Das AWS OpsWorks for Chef Automate Starterkit enthält ein Wrapper-Kochbuch`opsworks-audit`, das die richtige Version des Kochbuchs von Chef's Audit für Sie herunterlädt und installiert. Das `opsworks-audit` Kochbuch weist den `chef-client` Agenten außerdem an, Knoten anhand des DevSecSSH-Baseline-Profils zu bewerten, das Sie später in diesem Thema über die Compliance-Konsole von Chef installieren. Sie können Compliance unter Verwendung eines beliebigen Rezeptbuchs nach Ihren Anforderungen einrichten. Die Anweisungen in diesem Abschnitt verdeutlichen, wie Sie das `opsworks-audit`-Rezeptbuch implementieren.

So installieren Sie Compliance-Profile

1. Melden Sie sich beim web-basierten Dashboard von Chef Automate an, falls Sie dies noch nicht getan haben. Verwenden Sie bei der Erstellung Ihres Servers die Anmeldeinformationen, die Sie beim Herunterladen des Starter Kits erhalten haben. AWS OpsWorks for Chef Automate
2. Wählen Sie im Chef Automate-Dashboard die Registerkarte Compliance.



3. Wählen Sie in der linken Navigationsleiste Profile Store (Profilspeicher) und gehen Sie auf die Registerkarte Available (Verfügbar), um vordefinierte Profile anzuzeigen.
4. Durchsuchen Sie die Liste der Profile. Wählen Sie ein Profil, das mit dem Betriebssystem und der Konfiguration von mindestens einem Ihrer verwalteten Knoten übereinstimmt. Um Details über das Profil anzuzeigen, einschließlich einer Beschreibung der Verletzungen, auf die das Profil ausgelegt ist, und des zugrunde liegenden Regelcodes, wählen Sie **>** rechts neben dem Profileintrag. Sie können mehrere Profile auswählen.



5. Um die ausgewählten Profile auf Ihrem Chef Automate-Server zu installieren, wählen Sie Get (Abrufen).
6. Wenn der Download abgeschlossen ist, gehen Sie zum nächsten Verfahren über.

So installieren Sie das **`opsworks-audit`**-Rezeptbuch und richten es ein

1. Dieser Schritt ist optional, aber spart Zeit in Schritt 6, wenn Sie den Knotenausführungslisten Rezepte hinzufügen. Bearbeiten Sie die `roles/opsworks-example-role.rb`-Datei aus dem Starter Kit, das Sie bei der Erstellung Ihres AWS OpsWorks for Chef Automate -Servers heruntergeladen haben. Fügen Sie die folgende Zeilen hinzu. Die letzte Zeile ist auskommentiert, weil das Hinzufügen des `ssh-hardening`-Rezeptbuchs und des Rezepts für die Auflösung nichtkonformer Knoten nach der Ausführung Ihres Compliance-Scans optional ist.

```
run_list(
     "recipe[chef-client]",
     "recipe[apache2]",
     "recipe[opsworks-audit]"
     # "recipe[ssh-hardening]"
       )
```

2. Verwenden Sie einen Texteditor, um die gewünschten Rezeptbücher in Ihrer Berksfile-Datei anzugeben. Ein Beispiel für eine Berksfile-Datei finden Sie im Starter Kit. In diesem Beispiel

installieren wir das Chef Infra-Client(`chef-client`)-Rezeptbuch, das `apache2`-Rezeptbuch und das `opsworks-audit`-Rezeptbuch. Ihr Berksfile sollte etwa folgendermaßen aussehen.

```
source 'https://supermarket.chef.io
     cookbook 'chef-client'
     cookbook 'apache2', '~> 5.0.1'
     cookbook 'opsworks-audit', path: 'cookbooks/opsworks-audit', '~> 1.0.0'
```

Alle Rezeptbücher sind in der `metadata.rb`-Datei des Rezeptbuchs versioniert. Immer wenn Sie ein Rezeptbuch ändern, müssen Sie die Version des Rezeptbuchs ändern, die sich in seiner `metadata.rb` befindet.

3. Führen Sie den folgenden Befehl aus, um die Rezeptbücher in den Ordner `cookbooks` auf Ihrem lokalen oder auf Ihrem Arbeitscomputer herunterzuladen und zu installieren.

```
berks vendor cookbooks
```

4. Führen Sie den folgenden Befehl aus, um die von Anbietern bereitgestellten Rezeptbücher auf Ihren AWS OpsWorks for Chef Automate -Server hochzuladen.

```
knife upload .
```

5. Führen Sie den folgenden Befehl aus, um die Installation des `opsworks-audit`-Rezeptbuchs zu überprüfen, indem Sie eine Liste der Rezeptbücher anzeigen, die gegenwärtig auf dem Server verfügbar sind.

```
knife cookbook list
```

6. Fügen Sie Ihrem Server zu verwaltende Knoten hinzu, sofern Sie dies nicht bereits getan haben. Sie können die Zuordnung von Knoten automatisieren, indem Sie die Schritte in Automatisches Hinzufügen von Knoten AWS OpsWorks for Chef Automate ausführen, oder Knoten einzeln hinzufügen, indem Sie die Schritte in Fügen Sie Knoten einzeln hinzu ausführen. Bearbeiten Sie die Liste Ihrer Knoten, um die in Schritt 1 spezifizierte Rolle hinzuzufügen, `opsworks-example-role`. In diesem Beispiel bearbeiten wir das `RUN_LIST`-Attribut im `userdata`-Skript, das Sie verwenden, um die Zuordnung von Knoten zu automatisieren.

```
RUN_LIST="role[opsworks-example-role]"
```

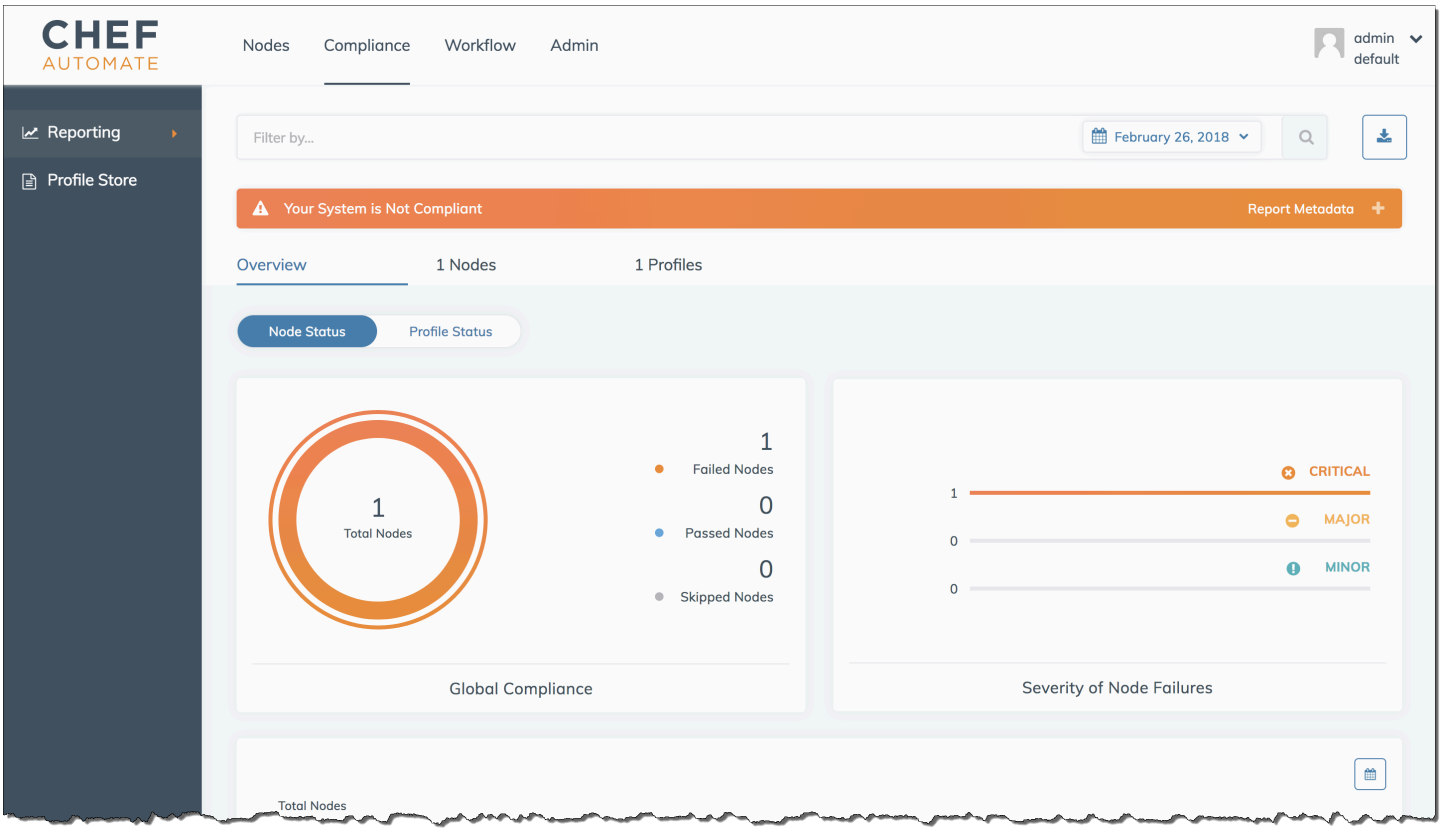
Wenn Sie Schritt 1 übersprungen und die Rolle nicht eingerichtet haben, fügen Sie der Ausführungsliste die Namen der einzelnen Rezepte hinzu. Speichern Sie Ihre Änderungen und folgen Sie den Schritten unter, Schritt 3: Erstellen von Instances mit einem unbeaufsichtigten Skript für die Zuordnung um Ihr Benutzerdatenskript auf Amazon EC2 EC2-Instances anzuwenden.

```
RUN_LIST="recipe[chef-client],recipe[apache2],recipe[opworks-audit]"
```

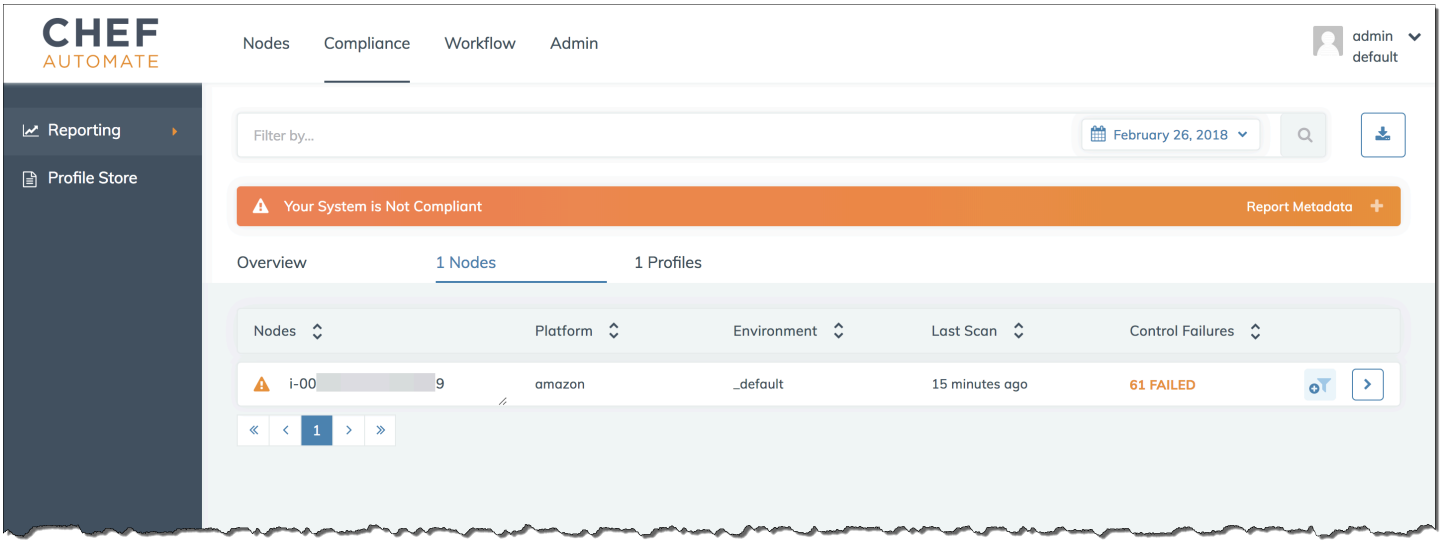
7. Nachdem Sie die Ausführungsliste für Ihre Knoten aktualisiert haben, führt der `chef-client`-Agent bei der nächsten Ausführung die von Ihnen spezifizierten Rezepte aus. Dies erfolgt standardmäßig alle 1800 Sekunden (30 Minuten). Nach der Ausführung werden Ihre Compliance-Ergebnisse auf dem Chef Automate-Dashboard angezeigt.

Ausführen eines Compliance-Scans

Sie sollten die Ergebnisse des Compliance-Scans im Dashboard von Chef Automate kurz nach der ersten Ausführung des Agenten-Daemons sehen, der stattfindet, nachdem Sie Knoten-Ausführungslisten konfiguriert haben.



Wählen Sie im Chef Automate-Dashboard die Registerkarte Compliance. Wählen Sie im linken Navigationsbereich die Option Reporting (Berichterstellung) aus. Wählen Sie auf die Registerkarte Profiles (Profile), dort Scan Results (Scan-Ergebnisse) und dann einen Knoten mit Scan-Fehlern aus, um mehr über die Regeln zu erfahren, gegen die ein Knoten verstoßen hat.



In der Regel werden fehlerhafte Scanergebnisse angezeigt, da neue Knoten noch nicht alle Regeln im DevSec SSH-Baseline-Profil erfüllen. Das DevSec Hardening Framework, ein Community-Projekt, bietet Kochbücher zur Behebung von Problemen, die gegen die Regeln im SSH-Baseline-Profil verstoßen. DevSec

(Optional) Auflösen nicht konformer Ergebnisse

Das Starterkit enthält ein Open-Source-Kochbuch, das Sie ausführen können**`ssh-hardening`**, um fehlerhafte Ergebnisse von Läufen gegen das SSH-Baseline-Profil zu korrigieren. DevSec

> **Note**
>
> Das **`ssh-hardening`** Cookbook nimmt Änderungen an Ihren Knoten vor, um den SSH-Baseline-Regeln zu entsprechen. DevSec Bevor Sie dieses Kochbuch auf Produktionsknoten ausführen, überprüfen Sie die Details zum DevSec SSH-Baseline-Profil in der Chef Automate-Konsole, um zu verstehen, auf welche Regelverstöße das Cookbook abzielt. Sehen Sie sich die Informationen über das als Open-Source bereitgestellte `ssh-hardening`-Rezeptbuch an, bevor Sie es auf Produktionsknoten ausführen.

So führen Sie das **`ssh-hardening`**-Rezeptbuch aus

1. Fügen Sie das `ssh-hardening`-Rezeptbuch in einem Texteditor Ihrer Berksfile-Datei hinzu. Ihr Berksfile sollte etwa folgendermaßen aussehen.

   ```
   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. Führen Sie die folgenden Befehle zum Herunterladen des `ssh-hardening`-Rezeptbuchs in Ihren lokalen Rezeptbuchordner aus, und laden Sie es dann auf Ihren AWS OpsWorks for Chef Automate -Server hoch.

   ```
   berks vendor cookbooks
   knife upload .
   ```

3. Fügen Sie Ihrer Knotenausführungsliste das `ssh-hardening`-Rezept hinzu, wie in den Schritten 1 und 6 von So installieren Sie das `opsworks-audit`-Rezeptbuch und richten es ein beschrieben.

   Wenn Sie die `opsworks-example-role.rb`-Datei aktualisieren, laden Sie Ihre Änderungen auf Ihren Server hoch, indem Sie den folgenden Befehl ausführen.

   ```
   knife upload .
   ```

   Wenn Sie die Ausführungsliste direkt aktualisieren, laden Sie Änderungen hoch, indem Sie den folgenden Befehl ausführen. Der Knotenname ist in der Regel die Instance-ID.

   ```
   knife node run_list add <node name> 'recipe[ssh-hardening]'
   ```

4. Da Sie das `chef-client`-Rezeptbuch verwenden, meldet sich Ihr Knoten in regelmäßigen Zeitabständen an (standardmäßig alle 30 Minuten). Beim nächsten Check-in wird das `ssh-hardening` Cookbook ausgeführt und trägt dazu bei, die Knotensicherheit zu verbessern, um die Regeln des DevSec SSH-Baseline-Profils zu erfüllen.

5. Warten Sie nach der ersten Ausführung des `ssh-hardening`-Rezeptbuchs 30 Minuten, bevor Sie wieder einen Compliance-Scan ausführen. Sehen Sie sich die Ergebnisse im Chef Automate-

Dashboard an. Die fehlerhaften Ergebnisse, die beim ersten Durchlauf des DevSec SSH-Baseline-Scans aufgetreten sind, sollten behoben werden.

Aktualisierungen der Compliance

Auf einem AWS OpsWorks for Chef Automate Server wird die Compliance-Funktionalität automatisch durch Ihre geplante Systemwartung aktualisiert. Sobald aktualisierte Versionen von Chef Automate, Chef Infra Server und Chef für Ihren AWS OpsWorks for Chef Automate Server verfügbar InSpec werden, müssen Sie möglicherweise die unterstützten Versionen des Audit-Kochbuchs und des InSpec Chef-Gems, die auf Ihrem Server laufen, überprüfen und aktualisieren. Profile, die Sie bereits auf Ihrem AWS OpsWorks for Chef Automate Server installiert haben, werden im Rahmen der Wartung nicht aktualisiert.

Community- und benutzerdefiniertes Compliance-Profile

Chef umfasst derzeit fast über 100 Compliance-Scan-Profile. Sie können der Liste Community- und benutzerdefinierten Profile hinzufügen und dann basierend auf diesen Profilen Compliance-Scans ausführen, genau wie für eingebundene Profile. Community-basierte Compliance-Profile stehen über den Chef Supermarket zur Verfügung. Benutzerdefinierte Profile sind auf Ruby basierende Programme, die einen Ordner mit Steuerelementen zur Spezifizierung Ihrer Scan-Regeln enthalten.

Weitere Informationen finden Sie unter:

- Blogbeitrag zur Ankündigung von Chef Compliance
- Online-Schulung zu Chef Automate Compliance
- InSpecChef-Webseite
- InSpec Anleitungen für Köche

Einen Knoten von einem AWS OpsWorks for Chef Automate Server trennen

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir

empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

In diesem Abschnitt wird beschrieben, wie Sie einen verwalteten Knoten von der Verwaltung durch einen Server trennen oder entfernen. AWS OpsWorks for Chef Automate Dieser Vorgang wird in der Befehlszeile ausgeführt. Sie können die Zuordnung von Knoten in der AWS OpsWorks for Chef Automate Managementkonsole nicht trennen. Derzeit erlaubt die AWS OpsWorks for Chef Automate API nicht, mehrere Knoten stapelweise zu entfernen. Mit dem in diesem Abschnitt verwendeten Befehl wird ein Knoten nach dem anderen getrennt.

Es empfiehlt sich, Knoten von Chef-Servern zu trennen, bevor Sie den Server löschen, damit die Knoten im weiteren Verlauf nicht ständig versuchen, sich erneut mit dem Server zu verbinden. Führen Sie dazu den `disassociate-node` AWS CLI Befehl aus.

So heben Sie die Zuordnung von Knoten auf

1. Führen Sie in der den folgenden Befehl aus AWS CLI, um die Zuordnung von Knoten zu trennen. *`Node_name`* ist der Name des Knotens, dessen Zuordnung Sie aufheben möchten. Für Amazon EC2 EC2-Instances ist dies die Instance-ID. *`Server_name`* ist der Name des Chef-Servers, von dem Sie den Knoten trennen möchten. `--engine-attributes` gibt Ihren `CHEF_AUTOMATE_ORGANIZATION`-Standardnamen an. Es werden alle drei dieser Parameter benötigt.

   Der Parameter `--region` wird nur benötigt, wenn Sie einen Knoten von einem Chef-Server außerhalb der Standardregion trennen möchten.

   ```
   aws opsworks-cm --region Region_name disassociate-node --node-
   name Node_name --server-name Server_name --engine-attributes
    "Name=CHEF_AUTOMATE_ORGANIZATION,Value='default'"
   ```

   Nachfolgend finden Sie einen Beispielbefehl.

   ```
   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. Warten Sie, bis in einer Antwortnachricht angezeigt wird, dass die Zuordnung aufgehoben wurde.

Nachdem Sie einen Knoten erfolgreich von einem AWS OpsWorks for Chef Automate Server getrennt haben, ist er möglicherweise immer noch im Chef Automate-Dashboard sichtbar. Standardmäßig erzwingt Chef einen Aufbewahrungszeitraum für Knotenzustandsinformationen und löscht den Knoten nach ein paar Tagen automatisch.

Weitere Informationen zum Löschen eines AWS OpsWorks for Chef Automate Servers finden Sie unterEinen AWS OpsWorks for Chef Automate Server löschen.

Verwandte Themen

Die folgenden AWS Blogbeiträge bieten weitere Informationen zur automatischen Zuordnung von Knoten zu Ihrem Chef Automate-Server, mithilfe von Auto Scaling Scaling-Gruppen oder innerhalb mehrerer Konten.

- Verwendung von AWS OpsWorks for Chef Automate zur Verwaltung von EC2-Instances mit Auto Scaling
- OpsWorks für Chef Automate — Automatisches Bootstrapping von Knoten in verschiedenen Konten

Einen AWS OpsWorks for Chef Automate Server löschen

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

In diesem Abschnitt wird beschrieben, wie Sie einen AWS OpsWorks for Chef Automate Server löschen. Durch Löschen eines Servers werden auch seine Ereignisse und Protokolle sowie alle auf dem Server gespeicherten Rezeptbücher gelöscht. Unterstützende Ressourcen (Amazon Elastic Compute Cloud-Instanz, Amazon Elastic Block Store-Volume usw.) werden zusammen mit allen automatisierten Backups ebenfalls gelöscht.

Obwohl durch Löschen eines Servers die Knoten nicht gelöscht werden, werden sie nicht mehr vom gelöschten Server verwaltet und versuchen fortlaufend, erneut eine Verbindung herzustellen. Aus diesem Grund empfehlen wir, die Zuordnung verwalteter Knoten aufzuheben, bevor Sie einen Chef-Server löschen. In dieser Version können Sie Knoten trennen, indem Sie einen AWS CLI Befehl ausführen.

Schritt 1: Aufheben der Zuordnung von verwalteten Knoten

Heben Sie die Zuordnung von Knoten zum Chef-Server auf, bevor Sie den Server löschen, damit die Knoten nicht fortlaufend versuchen, erneut eine Verbindung mit dem Server herzustellen. Führen Sie dazu den `disassociate-node` AWS CLI Befehl aus.

So heben Sie die Zuordnung von Knoten auf

1. Führen Sie in der den folgenden Befehl aus AWS CLI, um die Zuordnung von Knoten zu trennen. *`Server_name`* ist der Name des Chef-Servers, von dem Sie den Knoten trennen möchten.

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

2. Warten Sie, bis in einer Antwortnachricht angezeigt wird, dass die Zuordnung aufgehoben wurde.

Schritt 2: Löschen des Servers

1. Erweitern Sie auf dem Dashboard auf der Serverkachel das Menü Actions (Aktionen).
2. Wählen Sie Delete Server (Server löschen) aus.
3. Wenn Sie zum Bestätigen des Löschvorgangs aufgefordert werden, wählen Sie Yes (Ja) aus.

Zurücksetzen der Anmeldeinformationen für das Chef Automate-Dashboard

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Lebensende erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast,

kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

Sie sollten das Passwort, mit dem Sie sich beim Chef Automate-Dashboard anmelden, regelmäßig ändern. Sie können auch die in diesem Abschnitt gezeigten Amazon EC2 Systems Manager AWS CLI Manager-Befehle verwenden, um das Chef Automate-Dashboard-Passwort zu ändern, falls Sie es verloren haben. Welchen Befehl Sie verwenden, hängt davon ab, ob auf Ihrem Chef Automate-Server Version 1 oder Version 2 von Chef Automate ausgeführt wird.

1. Um die Instanz-ID Ihres Chef-Servers zurückzugeben, öffnen AWS Management Console Sie die folgende Seite.

   https://console.aws.amazon.com/ec2/v2/home?region = Region_des_Ihres Servers #Instanzen:Search= - Servername aws-opsworks-cm

   Für einen Chef-Server mit dem Namen MyChefServerin der Region USA West (Oregon) würde die Konsolen-URL beispielsweise wie folgt lauten.

   https://console.aws.amazon.com/ec2/v2/home?region=us-west-2#Instances:search = aws-opsworks-cm - MyChefServer

   Notieren Sie sich die Instance-ID, die in der Konsole angezeigt wird. Sie brauchen sie, um das Passwort zu ändern.

2. Um das Anmeldekennwort für das Chef Automate-Dashboard zurückzusetzen, führen Sie einen der folgenden AWS CLI Befehle aus, je nachdem, ob auf Ihrem Server Chef Automate 1 oder Chef Automate 2 ausgeführt wird. Ersetzen Sie *enterprise_name* durch Ihren Unternehmens- oder Organisationsnamen, *user_name* durch den Benutzernamen eines Administrators auf dem Server, *new_password durch das Passwort, das Sie verwenden möchten, und region_name* durch die Region, in der sich Ihr Server befindet. Wenn Sie keinen Unternehmensnamen angeben, erhält das Unternehmen den Namen `default`. Der Standardname für *enterprise_name* ist `default` (dieser Name wird für die Organisation immer bereitgestellt). *Erstellt für user_name nur einen Benutzer mit dem Namen.* AWS OpsWorks for Chef Automate `admin` Notieren Sie sich das neue Passwort und speichern Sie es an einem sicheren Speicherort.

   Für 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
```

Für 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. Warten Sie auf die Textausgabe (in diesem Fall der Befehl "ID") als Bestätigung, dass das Passwort geändert wurde.

AWS OpsWorks for Chef Automate API-Aufrufe protokollieren mit AWS CloudTrail

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

AWS OpsWorks for Chef Automate ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einer IAM-Identität oder einem AWS Dienst in ausgeführt werden. AWS OpsWorks for Chef Automate CloudTrail erfasst alle API-Aufrufe AWS OpsWorks for Chef Automate als Ereignisse, einschließlich Aufrufe von der AWS OpsWorks for Chef Automate Konsole und von Codeaufrufen an die AWS OpsWorks for Chef Automate APIs. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für AWS OpsWorks for Chef Automate. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen

können Sie die Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurde AWS OpsWorks for Chef Automate, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

AWS OpsWorks for Chef Automate Informationen in CloudTrail

CloudTrail ist in Ihrem AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn eine Aktivität in stattfindet AWS OpsWorks for Chef Automate, wird diese Aktivität zusammen mit anderen CloudTrail AWS Serviceereignissen im Ereignisverlauf in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS Konto ansehen, suchen und herunterladen. Weitere Informationen finden Sie unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich der Ereignisse für AWS OpsWorks for Chef Automate, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser standardmäßig für alle Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- Übersicht zum Erstellen eines Trails
- CloudTrail Unterstützte Dienste und Integrationen
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Alle AWS OpsWorks for Chef Automate Aktionen werden von der AWS OpsWorks for Chef Automate API-Referenz protokolliert CloudTrail und sind in dieser dokumentiert. Beispielsweise generieren Aufrufe der DescribeServersAktionen CreateServer, CreateBackup, und Einträge in den CloudTrail Protokolldateien.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Gibt an, ob die Anforderung mit Root- oder IAM-Benutzer-Anmeldeinformationen ausgeführt wurde.

- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter dem CloudTrail UserIdentity-Element.

Grundlegendes zu Einträgen AWS OpsWorks for Chef Automate in Protokolldateien

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag für die AWS OpsWorks for Chef Automate `CreateServer` Aktion.

```
{"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"
}
```

Problembehebung AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate hat am 5. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen bestehenden Kunden, zu Chef SaaS oder einer alternativen Lösung zu migrieren. Wenn du Fragen hast, kannst du dich auf AWS re:POST oder über den AWS Premium-Support an das AWS Support Team wenden.

Dieses Thema enthält einige häufig auftretende AWS OpsWorks for Chef Automate Probleme und Lösungsvorschläge für diese Probleme.

Themen

Allgemeine Tipps zur Problembehebung

Sollte es Ihnen nicht möglich sein, einen Chef-Server zu erstellen oder damit zu arbeiten, können Sie Fehlermeldungen oder Protokolle einsehen, die Ihnen helfen, den Fehler zu beheben. Die folgenden Aufgaben beschreiben allgemeine Ausgangspunkte bei der Fehlerbehebung eines Chef-Server Problems. Weitere Informationen zu bestimmten Fehlern und Lösungen finden Sie im Abschnitt Behebung bestimmter Fehler dieses Themas.

- Verwenden Sie die AWS OpsWorks for Chef Automate Konsole, um Fehlermeldungen anzuzeigen, wenn ein Chef-Server nicht gestartet werden kann. Fehlermeldungen im Zusammenhang mit dem Starten und Ausführen des Servers werden auf der Detailseite des Chef-Servers oben

angezeigt. Fehler können von AWS OpsWorks for Chef Automate AWS CloudFormation, oder Amazon EC2-Diensten herrühren, die zum Erstellen eines Chef-Servers verwendet werden. Auf der Detailseite können Sie auch Ereignisse sehen, die auf einem laufenden Server auftreten und Fehlerereignismeldungen beinhalten können.

- Zur Lösung der EC2-Probleme können Sie eine Verbindung zu Ihrer Server-Instance über SSH herstellen und die Protokolle überprüfen. EC2-Instance-Protokolle werden im `/var/log/aws/opsworks-cm`-Verzeichnis gespeichert. Diese Protokolle erfassen Befehlsausgaben beim Starten AWS OpsWorks for Chef Automate eines Chef-Servers.

Behebung bestimmter Fehler

Themen

Der Server befindet sich in einem Zustand „Verbindung verloren“

Problem: Der Status eines Servers wird als Verbindung unterbrochen angezeigt.

Ursache: Dies tritt am häufigsten auf, wenn eine Entität außerhalb von Änderungen an einem AWS OpsWorks for Chef Automate Server oder dessen unterstützenden Ressourcen AWS OpsWorks vornimmt. AWS OpsWorks kann im Status „Verbindung verloren“ keine Verbindung zu Chef Automate-Servern herstellen, um Wartungsaufgaben wie das Erstellen von Backups, das Anwenden

von Betriebssystem-Patches oder das Aktualisieren von Chef Automate zu erledigen. Infolgedessen fehlen auf Ihrem Server möglicherweise wichtige Updates, er ist anfällig für Sicherheitsprobleme oder er funktioniert auf andere Weise nicht wie erwartet.

Lösung: Führen Sie die folgenden Schritte aus, um die Serververbindung wiederherzustellen.

1. Stellen Sie sicher, dass Ihre Servicerolle über alle erforderlichen Berechtigungen verfügt.

    a. Wählen Sie auf der Seite Einstellungen für Ihren Server unter Netzwerk und Sicherheit den Link für die Servicerolle aus, die der Server verwendet. Dadurch wird die Servicerolle zur Anzeige in der IAM-Konsole geöffnet.

    b. Vergewissern Sie sich, dass auf der Registerkarte Berechtigungen der Eintrag in der Liste der Berechtigungsrichtlinien aufgeführt `AWSOpsWorksCMServiceRole` ist. Wenn sie nicht aufgeführt ist, fügen Sie die `AWSOpsWorksCMServiceRole` verwaltete Richtlinie manuell zur Rolle hinzu.

    c. Stellen Sie auf der Registerkarte Vertrauensbeziehungen sicher, dass die Servicerolle über eine Vertrauensrichtlinie verfügt, die darauf vertraut, dass der `opsworks-cm.amazonaws.com` Dienst Rollen in Ihrem Namen übernimmt. Weitere Informationen zur Verwendung von Vertrauensrichtlinien mit Rollen finden Sie unter Ändern einer Rolle (Konsole) oder im AWS Sicherheits-Blogbeitrag So verwenden Sie Vertrauensrichtlinien mit IAM-Rollen.

2. Stellen Sie sicher, dass Ihr Instanzprofil über alle erforderlichen Berechtigungen verfügt.

    a. Wählen Sie auf der Seite Einstellungen für Ihren Server unter Netzwerk und Sicherheit den Link für das Instanzprofil aus, das der Server verwendet. Dadurch wird das Instanzprofil zur Anzeige in der IAM-Konsole geöffnet.

    b. Stellen Sie auf der Registerkarte Berechtigungen sicher, dass `AmazonEC2RoleforSSM` beide in der Liste der Berechtigungsrichtlinien aufgeführt `AWSOpsWorksCMInstanceProfileRole` sind. Wenn eine oder beide nicht aufgeführt sind, fügen Sie diese verwalteten Richtlinien manuell zur Rolle hinzu.

    c. Vergewissern Sie sich auf der Registerkarte Vertrauensbeziehungen, dass die Servicerolle über eine Vertrauensrichtlinie verfügt, die darauf vertraut, dass der `ec2.amazonaws.com` Dienst Rollen in Ihrem Namen übernimmt. Weitere Informationen zur Verwendung von Vertrauensrichtlinien mit Rollen finden Sie unter Ändern einer Rolle (Konsole) oder im AWS Sicherheits-Blogbeitrag So verwenden Sie Vertrauensrichtlinien mit IAM-Rollen.

3. Stellen Sie in der Amazon EC2 EC2-Konsole sicher, dass Sie sich in derselben Region wie die Region des AWS OpsWorks for Chef Automate Servers befinden, und starten Sie dann die EC2-Instance neu, die Ihr Server verwendet.

   a. *Wählen Sie die EC2-Instance mit dem Namen Servername aus. aws-opsworks-cm-instance-*

   b. Wählen Sie im Menü Instanzstatus die Option Reboot instance aus.

   c. Warten Sie bis zu 15 Minuten, bis Ihr Server neu gestartet und vollständig online ist.

4. Stellen Sie in der AWS OpsWorks for Chef Automate Konsole auf der Seite mit den Serverdetails sicher, dass der Serverstatus jetzt fehlerfrei ist.

Wenn der Serverstatus nach Durchführung der vorherigen Schritte immer noch Verbindung verloren lautet, versuchen Sie es mit einer der folgenden Methoden.

- Ersetzen Sie den Server, indem Sie einen neuen Server erstellen und das Original löschen. Wenn Daten auf dem aktuellen Server für Sie wichtig sind, stellen Sie den Server anhand einer aktuellen Sicherung wieder her und überprüfen Sie, ob die Daten auf dem neuesten Stand sind, bevor Sie den ursprünglichen Server löschen, der nicht mehr reagiert.
- Wenden Sie sich an den AWS Support.

Verwaltete Knoten im Chef Automate-Dashboard in der Spalte für fehlende Einträge

Problem: Ein verwalteter Knoten erscheint in der Chef Automate-Dashboard-Spalte Missing (Fehlend).

Ursache: Wenn ein Knoten sich länger als 12 Stunden nicht mit dem Chef Automate-Server verbunden hat und `chef-client` nicht auf dem Knoten ausgeführt werden kann, ändert sich der Status des Knotens zurück in den Status, der vor den 12 Stunden aktiv war, und der Knoten wird in die Spalte Missing (Fehlend) des Chef Automate-Dashboard verschoben.

Lösung: Überprüfen Sie, ob der Knoten online ist. Versuchen Sie, `knife node show` *`node_name`* `--run-list` auszuführen, um zu sehen, ob `chef-client` auf dem Knoten ausgeführt werden kann, oder `knife node show -l` *`node_name`*, um alle Informationen über den Knoten anzuzeigen. Der Knoten könnte offline oder vom Netzwerk abgemeldet sein.

Chef-Tresor kann nicht erstellt werden; der Befehl **`knife vault`** schlägt fehl

Problem: Sie versuchen, einen Tresor auf Ihrem Chef Automate-Server zu erstellen (z. B. einen Tresor zum Speichern von Anmeldeinformationen für die Domäneneinbindung von Windows-basierten Knoten), indem Sie den Befehl `knife vault` ausführen. Der Befehl gibt eine ähnliche Fehlermeldung wie die diese zurück.

```
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!
```

Der pivotale Benutzer wird nicht zurückgegeben, wenn Sie `knife user list` remote ausgeführt haben. Sie können jedoch den pivotalen Benutzer in den Ergebnissen sehen, wenn Sie den Befehl `chef-server-ctl user-show` lokal auf Ihrem Chef Automate-Server ausführen. Mit anderen Worten: Ihr Befehl `knife vault` kann den pivotalen Benutzer nicht finden, aber Sie wissen, dass er vorhanden ist.

Ursache: Obwohl der pivotale Benutzer in Chef als Superuser gilt und über vollständige Berechtigungen verfügt, ist er nicht Mitglied irgendeiner Organisation, einschließlich der `default`-Organisation, die in AWS OpsWorks for Chef Automate verwendet wird. Der Befehl `knife user list` gibt alle Benutzer zurück, die sich in der aktuellen Organisation in Ihrer Chef-Konfiguration befinden. Der Befehl `chef-server-ctl user-show` gibt alle Benutzer zurück, unabhängig von der Organisation und einschließlich des pivotalen Benutzers.

Lösung: Um das Problem zu beheben, fügen Sie den pivotalen Benutzer der Standard-Organisation hinzu, indem Sie `knife opc` ausführen.

Zunächst müssen Sie das Plug-In knife opc installieren.

```
chef gem install knife-opc
```

Nach der Installation des Plug-Ins führen Sie den folgenden Befehl aus, um den pivotalen Benutzer der Standard-Organisation hinzuzufügen.

```
knife opc org user add default pivotal
```

Sie können überprüfen, ob der pivotale Benutzer Teil der Standard-Organisation ist, indem Sie `knife user list` erneut ausführen. `pivotal` sollte in den Ergebnissen aufgeführt sein. Starten Sie anschließend `knife vault` erneut.

Servererstellung schlägt mit der Nachricht "Die angefragte Konfiguration wird derzeit nicht unterstützt" fehl

Problem: Sie versuchen einen Chef Automate-Server zu erstellen. Die Servererstellung schlägt jedoch mit einer Fehlermeldung wie "Die angefragte Konfiguration wird derzeit nicht unterstützt" fehl. Bitte überprüfen Sie die Dokumentation auf unterstützte Konfigurationen.

Ursache: Ein nicht unterstützter Instance-Typ könnte für den Chef Automate-Server angegeben worden sein. Wenn Sie einen Chef Automate-Server in einer VPC erstellen möchten, die über eine nicht standardmäßige Tenancy verfügt, z. B. eine für dedizierte Instances, müssen alle Instances innerhalb der angegebenen VPC auch einer dedizierten oder Host-Tenancy angehören. Da einige Instance-Typen, z. B. t2, nur mit einem Standard-Abonnement verfügbar sind, könnte der Chef Automate-Server-Instance-Typ möglicherweise nicht durch die angegebene VPC unterstützt werden. Außerdem schlägt die Servererstellung fehl.

Lösung: Wenn Sie eine VPC mit einer Nicht-Standard-Tenancy auswählen, nutzen Sie einen m4-Instance-Typ, der eine dedizierte Tenancy unterstützt.

Der Chef-Server erkennt keine Organisationsbezeichnungen, die dem Chef Automate-Dashboard hinzugefügt wurden.

Problem: Sie haben dem Chef Automate-Dashboard neue Workflow-Organisationsnamen hinzugefügt oder einen anderen `CHEF_AUTOMATE_ORGANIZATION`-Wert als `"default"` im unbeaufsichtigten Knoten-Zuordnungsskript angegeben. Die Knotenzuordnung schlägt jedoch fehl. Ihr AWS OpsWorks for Chef Automate -Server erkennt die neuen Organisationsnamen nicht.

Ursache: Workflow-Organisationsnamen und Chef-Server-Organisationsnamen sind nicht identisch. Sie können neue Workflow-Organisationen in dem webbasierten Chef Automate-Dashboard erstellen, aber keine Chef-Server-Organisationsnamen. Sie können das Chef Automate-Dashboard nur nutzen, um vorhandene Chef-Server-Organisationen anzuzeigen. Eine neue Organisation, die Sie in dem Chef Automate-Dashboard erstellen, ist eine Workflow-Organisation und wird von dem Chef-Server nicht erkannt. Sie können keine neuen Organisationsnamen erstellen, indem Sie sie in dem Knoten-Zuordnungsskript angeben. Der Verweis auf einen Organisationsnamen in einem Knoten-Zuordnungsskript bewirkt, dass die Knotenzuordnung fehlschlägt, falls die Organisation nicht zuerst dem Chef-Server hinzugefügt wurde.

Lösung: Um neue Organisationen zu erstellen, die auf dem Chef-Server erkannt werden, verwenden Sie den Befehl `knife opc org create` oder führen Sie `chef-server-ctl org-create` aus.

Die Amazon EC2 EC2-Instance des Servers konnte nicht erstellt werden

Problem: Die Servererstellung schlug mit einer ähnlichen Fehlermeldung wie dieser fehl: "Die folgende Ressource(n) konnte(n) nicht erstellt werden: [EC2Instance]. Fehler beim Empfang von 1 Ressource-Signal innerhalb der angegebenen Dauer."

Ursache: Dies ist am wahrscheinlichsten, da die EC2-Instance keinen Zugriff auf das Netzwerk hat.

Lösung: Stellen Sie sicher, dass die Instance über einen ausgehenden Internetzugang verfügt und der AWS Service-Agent Befehle ausgeben kann. Stellen Sie sicher, dass Ihre VPC (eine VPC mit einem einzigen öffentlichen Subnetz) DNS resolution (DNS-Auflösung) aktiviert hat und Ihr Subnetz die Einstellung Auto-assign Public IP (Öffentliche IP-Adresse automatisch zuweisen) aktiviert hat.

Service-Rollen-Fehler verhindert die Servererstellung

Problem: Die Servererstellung schlägt fehl und es wird eine Fehlermeldung angezeigt, die besagt: „Nicht autorisiert, sts auszuführen:AssumeRole.“

Ursache: Dies kann auftreten, wenn die Service-Rolle, die Sie nutzen, nicht über die erforderlichen Berechtigungen zum Erstellen eines neuen Servers verfügt.

Lösung: Öffnen Sie die AWS OpsWorks for Chef Automate Konsole und verwenden Sie die Konsole, um eine neue Servicerolle und eine Instanzprofilrolle zu generieren. Wenn Sie lieber Ihre eigene Servicerolle verwenden möchten, fügen Sie die AWSOpsWorksCMServiceRoleRichtlinie der Rolle hinzu. Vergewissern Sie sich, dass opsworks-cm.amazonaws.com unter den Diensten in den Vertrauensbeziehungen der Rolle aufgeführt ist. Stellen Sie sicher, dass der Servicerolle, die dem Chef-Server zugeordnet ist, die verwaltete Richtlinie angehängt ist. AWSOpsWorksCMServiceRole

Elastisches Limit der IP-Adresse überschritten

Problem: Servererstellung schlägt fehl mit folgender Fehlermeldung: "Die folgende Ressource(n) konnte(n) nicht erstellt werden: [EIP, EC2Instance]. Ressourcenerstellung abgebrochen, die maximale Anzahl an Adressen wurde erreicht."

Ursache: Dieses Problem tritt auf, wenn Ihr Konto die maximale Anzahl an Elastic IP (EIP)-Adressen genutzt hat. Das standardmäßige EIP-Adressenlimit ist fünf.

Lösung: Sie können entweder bestehende EIP-Adressen freigeben oder solche löschen, die Ihr Konto nicht aktiv verwendet, oder Sie können sich an den AWS Kundensupport wenden, um das Limit an EIP-Adressen zu erhöhen, das mit Ihrem Konto verknüpft ist.

Anmeldung beim Chef Automate-Dashboard nicht möglich

Problem: Das Chef Automate-Dashboard zeigt eine ähnliche Fehlermeldung wie diese an: "Ursprungsübergreifende Anfrage blockiert: Die gleiche Ursprungsrichtlinie verbietet das Lesen der Remote-Ressource auf https://myserver-name.region.opsworks-cm.io/api/v0/e/default/verify-token. (Grund: CORS-Header "Access-Control-Allow-Origin" fehlt)". Der Fehler kann folgendermaßen aussehen: "Die eingegebene Benutzer-ID/Passwort-Kombination ist falsch."

Ursache: Das Chef Automate-Dashboard legt ausdrücklich den FQDN fest und akzeptiert keine relativen URLs. Derzeit ist es nicht möglich, sich über die Chef-Server-IP-Adresse anzumelden. Sie können sich nur anmelden, indem Sie den DNS-Namen des Servers nutzen.

Lösung: Melden Sie sich beim Chef Automate-Dashboard nur an, indem Sie den DNS-Namenseintrag des Servers nutzen und nicht seine IP-Adresse. Sie können auch versuchen, die Chef Automate-Dashboard-Anmeldeinformationen zurückzusetzen, indem Sie einen AWS CLI -Befehl ausführen, der in Zurücksetzen der Anmeldeinformationen für das Chef Automate-Dashboard beschrieben wird.

Unbeaufsichtigte Knotenzuordnung fehlgeschlagen

Problem: Die unbeaufsichtigte oder automatische Zuordnung neuer Amazon EC2 EC2-Knoten schlägt fehl. Knoten, die dem Chef-Server hinzugefügt werden sollten, erscheinen nicht im Chef Automate-Dashboard und sind nicht in den Ergebnisse der Befehle `knife client show` oder `knife node show` aufgeführt.

Ursache: Dies kann auftreten, wenn Sie nicht über eine IAM-Rolle verfügen, die als Instance-Profil eingerichtet wurde und die gestattet, dass `opsworks-cm`-API-Aufrufe mit neuen EC2-Instances kommunizieren können.

Lösung: Fügen Sie Ihrem EC2-Instance-Profil eine Richtlinie an, die es erlaubt, dass die API-Aufrufe `AssociateNode` und `DescribeNodeAssociationStatus` mit EC2 zusammenarbeiten können, wie in Automatisches Hinzufügen von Knoten AWS OpsWorks for Chef Automate beschrieben.

Die Systemwartung schlägt fehl

AWS OpsWorks CM führt wöchentliche Systemwartungen durch, um sicherzustellen, dass die neuesten Nebenversionen von Chef Server und Chef Automate Server, einschließlich

Sicherheitsupdates, immer auf einem AWS OpsWorks for Chef Automate-Server laufen. Wenn die Systemwartung aus irgendeinem Grund fehlschlägt, werden Sie über den Fehler AWS OpsWorks CM informiert. Weitere Hinweise zur Systemwartung finden Sie unterSystemwartung in AWS OpsWorks for Chef Automate.

In diesem Abschnitt werden mögliche Fehlerursachen beschrieben und Lösungen vorgeschlagen.

Themen

- Ein Fehler im Servicerollen- oder Instanzprofil verhindert die Systemwartung

Ein Fehler im Servicerollen- oder Instanzprofil verhindert die Systemwartung

Problem: Die Systemwartung schlägt fehl und es wird eine Fehlermeldung angezeigt, die besagt, dass Sie nicht berechtigt sind, sts auszuführen:AssumeRole, oder es wird eine ähnliche Fehlermeldung zu den Berechtigungen angezeigt.

Ursache: Dieses Problem kann auftreten, wenn entweder die von Ihnen verwendete Servicerolle oder das Instanzprofil nicht über ausreichende Berechtigungen für die Durchführung der Systemwartung auf dem Server verfügt.

Lösung: Stellen Sie sicher, dass Ihre Servicerolle und Ihr Instanzprofil über alle erforderlichen Berechtigungen verfügen.

1. Stellen Sie sicher, dass Ihre Servicerolle über alle erforderlichen Berechtigungen verfügt.

    a. Wählen Sie auf der Seite Einstellungen für Ihren Server unter Netzwerk und Sicherheit den Link für die Servicerolle aus, die der Server verwendet. Dadurch wird die Servicerolle zur Anzeige in der IAM-Konsole geöffnet.

    b. Vergewissern Sie sich auf der Registerkarte „Berechtigungen“, dass sie der Servicerolle zugeordnet `AWSOpsWorksCMServiceRole` ist. Wenn sie nicht aufgeführt `AWSOpsWorksCMServiceRole` ist, fügen Sie diese Richtlinie der Rolle hinzu.

    c. Vergewissern Sie sich, dass opsworks-cm.amazonaws.com unter den Diensten in den Vertrauensbeziehungen der Rolle aufgeführt ist. Weitere Informationen zur Verwendung von Vertrauensrichtlinien mit Rollen finden Sie unter Rolle ändern (Konsole) oder im AWS Sicherheits-Blogbeitrag So verwenden Sie Vertrauensrichtlinien mit IAM-Rollen.

2. Stellen Sie sicher, dass Ihr Instanzprofil über alle erforderlichen Berechtigungen verfügt.

a. Wählen Sie auf der Seite Einstellungen für Ihren Server unter Netzwerk und Sicherheit den Link für das Instanzprofil aus, das der Server verwendet. Dadurch wird das Instanzprofil zur Anzeige in der IAM-Konsole geöffnet.

b. Stellen Sie auf der Registerkarte Berechtigungen sicher, dass `AmazonEC2RoleforSSM` beide in der Liste der Berechtigungsrichtlinien aufgeführt `AWSOpsWorksCMInstanceProfileRole` sind. Wenn eine oder beide nicht aufgeführt sind, fügen Sie diese verwalteten Richtlinien manuell zur Rolle hinzu.

c. Vergewissern Sie sich auf der Registerkarte Vertrauensbeziehungen, dass die Servicerolle über eine Vertrauensrichtlinie verfügt, die darauf vertraut, dass der `ec2.amazonaws.com` Dienst Rollen in Ihrem Namen übernimmt. Weitere Informationen zur Verwendung von Vertrauensrichtlinien mit Rollen finden Sie unter Ändern einer Rolle (Konsole) oder im AWS Sicherheits-Blogbeitrag So verwenden Sie Vertrauensrichtlinien mit IAM-Rollen.

Weitere Hilfe und Support

Wenn Ihr spezifisches Problem in diesem Thema nicht beschrieben wird oder Sie die Vorschläge in diesem Thema ausprobiert und weiterhin Probleme haben, besuchen Sie die AWS OpsWorks -Foren.

Sie können auch das AWS Support-Center besuchen. Das AWS Support Center ist die zentrale Anlaufstelle für die Erstellung und Verwaltung von AWS Support-Fällen. Das AWS Support Center enthält auch Links zu anderen hilfreichen Ressourcen wie Foren, technischen FAQs, Servicestatus und AWS Trusted Advisor.

Sicherheit in AWS OpsWorks Configuration Management (CM)

Die Sicherheit in der Cloud hat für AWS höchste Priorität. Als AWS-Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die zur Erfüllung der Anforderungen von Organisationen entwickelt wurden, für die Sicherheit eine kritische Bedeutung hat.

Sicherheit gilt zwischen AWS und Ihnen eine geteilte Verantwortung. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS-Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS-Compliance-Programme regelmäßig. Informationen zu den Compliance-Programmen, die für AWS OpsWorks CM gelten, finden Sie unter Vom Compliance-Programm abgedeckte AWS-Services.
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Dienst bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

In dieser Dokumentation wird erläutert, wie das Modell der übergreifenden Verantwortlichkeit bei der Verwendung von AWS OpsWorks CM zum Tragen kommt. Die folgenden Themen zeigen Ihnen, wie Sie AWS OpsWorks CM zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie erfahren auch, wie Sie andere AWS-Services nutzen können, die Ihnen helfen, Ihre AWS OpsWorks CM-Ressourcen zu überwachen und zu sichern.

Themen

- Infrastruktursicherheit in AWS OpsWorks CM
- Konfigurations- und Schwachstellenanalyse in AWS OpsWorks CM
- Bewährte Sicherheitsmethoden für AWS OpsWorksCM

Datenschutz in AWS OpsWorks CM

Das AWS Modell der gilt für den Datenschutz in AWS OpsWorks Configuration Management. Wie in diesem Modell beschrieben, ist AWS für den Schutz der globalen Infrastruktur verantwortlich, in der die gesamte AWS Cloud ausgeführt wird. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS-Modell der geteilten Verantwortung und in der DSGVO im AWS-Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, AWS-Konto-Anmeldeinformationen zu schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor Authentifizierung (MFA).
- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit AWS CloudTrail ein.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit

OpsWorks CM oder anderen zusammenarbeiten und die KonsoleAWS CLI, API oder AWS SDKs AWS-Services verwenden. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Die Namen der OpsWorks CM-Server sind nicht verschlüsselt.

OpsWorks CM sammelt im Zuge der Erstellung und Wartung Ihrer AWS OpsWorks for Chef Automate und AWS OpsWorks for Puppet Enterprise Server die folgenden Kundendaten.

- Denn OpsWorks für Puppet Enterprise sammeln wir private Schlüssel, die Puppet Enterprise verwendet, um die Kommunikation zwischen Ihrem Puppet-Master und verwalteten Knoten zu ermöglichen.
- Für AWS OpsWorks for Chef Automate sammeln wir private Schlüssel für Zertifikate, die Sie an den Service anhängen, wenn Sie eine benutzerdefinierte Domäne verwenden. Der private Schlüssel, den Sie angeben, wenn Sie einen Chef Automate-Server mit einer benutzerdefinierten Domäne erstellen, wird an Ihren Server weitergeleitet.

OpsWorks CM-Server speichern Ihren Konfigurationscode, z. B. Chef-Kochbücher oder Puppet Enterprise-Module. Obwohl dieser Code in Serversicherungen gespeichert ist, hat AWS keinen Zugriff darauf. Dieser Inhalt ist verschlüsselt, und nur Administratoren in Ihrem AWS-Konto können darauf zugreifen. Wir empfehlen Ihnen, Ihren Chef- oder Puppet-Konfigurationscode mithilfe der empfohlenen Protokolle für Ihre Quell-Repositorys zu sichern. Sie können beispielsweise die Berechtigungen auf Repositorys in AWS CodeCommit beschränken oder die Richtlinien auf der GitHub Website zur Sicherung von Repositorys befolgen. GitHub

OpsWorks CM verwendet keine vom Kunden bereitgestellten Inhalte, um den Service aufrechtzuerhalten oder Kundenprotokolle zu führen. Protokolle über Ihre OpsWorks CM-Server werden in Ihrem Konto in Amazon S3 S3-Buckets gespeichert. IP-Adressen von Benutzern, die eine Verbindung zu Ihren OpsWorks CM-Servern herstellen, werden von AWS protokolliert.

Integration in AWS Secrets Manager

Ab dem 3. Mai 2021, wenn Sie einen neuen Server in OpsWorks CM erstellen, speichert OpsWorks CM Geheimnisse für den Server inAWS Secrets Manager. Für neue Server werden die folgenden Attribute als Secrets in Secrets Manager gespeichert.

- Chef Automate Server
  - Privater HTTPS-Schlüssel (nur Server, die keine benutzerdefinierte Domain verwenden)
  - Das Administratorkennwort von Chef Automate (CHEF_AUTOMATE_ADMIN_PASSWORD)
- Meister von Puppet Enterprise
  - Privater HTTPS-Schlüssel (nur Server, die keine benutzerdefinierte Domäne verwenden)
  - Puppet-Administratorkennwort (PUPPET_ADMIN_PASSWORD)
  - Puppet R10k-Fernbedienung (PUPPET_R10K_REMOTE)

Für bestehende Server, die keine benutzerdefinierte Domäne verwenden, ist das einzige in Secrets Manager gespeicherte Geheimnis, sowohl für Chef Automate- als auch für Puppet Enterprise-Server, der private HTTPS-Schlüssel, da dieser während der automatischen, wöchentlichen Systemwartung generiert wird.

OpsWorks CM speichert Geheimnisse automatisch in Secrets Manager, und dieses Verhalten ist nicht vom Benutzer konfigurierbar.

Datenverschlüsselung

AWS OpsWorks CM verschlüsselt Serversicherungen und die Kommunikation zwischen autorisierten AWS-Benutzern und ihren AWS OpsWorks CM-Servern. Die Amazon EBS-Root-Volumes von AWS OpsWorks CM-Servern sind jedoch nicht verschlüsselt.

Verschlüsselung im Ruhezustand

AWS OpsWorks CM-Server-Sicherungen werden verschlüsselt. Die Amazon EBS-Root-Volumes von AWS OpsWorks CM-Servern sind jedoch nicht verschlüsselt. Dies ist nicht vom Benutzer konfigurierbar.

Verschlüsselung während der Übertragung

AWS OpsWorksCM verwendet HTTP mit TLS-Verschlüsselung. AWS OpsWorks CM verwendet standardmäßig selbstsignierte Zertifikate für die Bereitstellung und Verwaltung von Servern, wenn kein signiertes Zertifikat von Benutzern bereitgestellt wird. Es wird empfohlen, ein Zertifikat zu verwenden, das von einer Zertifizierungsstelle (Certificate Authority, CA) signiert wurde.

Schlüsselverwaltung

Kundenverwaltete AWS Key Management Service-Schlüssel und von AWS verwaltete Schlüssel werden derzeit nicht von AWS OpsWorks CM unterstützt.

Identity and Access Management für AWS OpsWorks CM

AWS Identity and Access Management (IAM) ist ein AWS Dienst, der einem Administrator hilft, den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um CM-Ressourcen zu verwenden OpsWorks . IAM ist ein AWS Dienst, den Sie ohne zusätzliche Kosten nutzen können.

Themen

Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in OpsWorks CM ausführen.

Dienstbenutzer — Wenn Sie den OpsWorks CM-Dienst für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Da Sie für Ihre Arbeit mehr OpsWorks CM-Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Fuktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern müssen. Wenn Sie in OpsWorks CM nicht auf eine Funktion zugreifen können, finden Sie weitere Informationen unterProblembehandlung bei AWS OpsWorks CM Identity and Access.

Serviceadministrator — Wenn Sie in Ihrem Unternehmen für OpsWorks CM-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf OpsWorks CM. Es ist Ihre Aufgabe, zu bestimmen, auf welche OpsWorks CM-Funktionen und Ressourcen Ihre Servicebenutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM mit OpsWorks CM nutzen kann, finden Sie unterSo funktioniert AWS OpsWorks CM mit IAM.

IAM-Administrator — Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien zur Verwaltung des Zugriffs auf CM schreiben können. OpsWorks Beispiele für identitätsbasierte OpsWorks CM-Richtlinien, die Sie in IAM verwenden können, finden Sie unter. AWS OpsWorks Beispiele für identitätsbasierte CM-Richtlinien

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS , übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter So melden Sie sich bei Ihrem an AWS-Konto im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu signieren, finden Sie im IAM-Benutzerhandbuch unter AWS API-Anfragen signieren.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS Empfiehlt beispielsweise, die Multi-Faktor-

Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center - Benutzerhandbuch und Verwenden der Multi-Faktor-Authentifizierung (MFA) in AWS im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität innerhalb von Ihnen AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Erstellen eines IAM-Benutzers (anstatt einer Rolle) im IAM-Benutzerhandbuch.

> **Warning**
>
> IAM-Benutzer verfügen über langfristige Anmeldeinformationen, was ein Sicherheitsrisiko darstellt. Um dieses Risiko zu minimieren, empfehlen wir, diesen Benutzern nur die Berechtigungen zu gewähren, die sie für die Ausführung der Aufgabe benötigen, und diese Benutzer zu entfernen, wenn sie nicht mehr benötigt werden.

IAM-Rollen

Eine IAM-Rolle ist eine Identität innerhalb Ihres Unternehmens, für AWS-Konto die bestimmte Berechtigungen gelten. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die Rollen wechseln. Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter Verwenden von IAM-Rollen im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter Erstellen von Rollen für externe Identitätsanbieter im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center -Benutzerhandbuch.
- Temporäre IAM-Benutzerberechtigungen – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen

Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie unter Kontenübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

- Serviceübergreifender Zugriff — Einige verwenden Funktionen in anderen. AWS-Services AWS-Services Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon-EC2 aus oder speichert Objekte in Amazon-S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Forward Access Sessions (FAS) — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
  - Servicerolle – Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.
  - Dienstbezogene Rolle — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- Anwendungen, die auf Amazon EC2 ausgeführt werden — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen AWS CLI . AWS Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter Erstellen einer IAM-Rolle (anstatt eines Benutzers) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie

mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter Auswahl zwischen verwalteten und eingebundenen Richtlinien im IAM-Benutzerhandbuch.

OpsWorks CM unterstützt benutzerdefinierte Richtlinien, die Sie in IAM erstellen und Benutzern, Rollen oder Gruppen zuordnen.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

OpsWorks CM unterstützt keine ressourcenbasierten Richtlinien.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. AWS WAF Weitere Informationen“ zu ACLs finden Sie unter Zugriffskontrollliste (ACL) – Übersicht (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

OpsWorks CM verwendet keine ACLs.

Weitere Richtlinientypen

OpsWorks CM unterstützt die folgenden anderen Richtlinientypen nicht.

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen — Eine Berechtigungsgrenze ist eine erweiterte Funktion, mit der Sie die maximalen Berechtigungen festlegen, die eine identitätsbasierte Richtlinie einer IAM-Entität (Benutzer oder Rolle) gewähren kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die resultierenden Berechtigungen sind eine Schnittmenge der identitätsbasierten Richtlinien der Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.
- Service Control Policies (SCPs) — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer AWS Konten, die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Das SCP beschränkt die Berechtigungen für Entitäten in Mitgliedskonten, einschließlich der einzelnen Konten. Root-Benutzer des AWS-Kontos Weitere Informationen über Organizations und SCPs finden Sie unter Funktionsweise von SCPs im AWS Organizations - Leitfaden.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter Bewertungslogik für Richtlinien.

So funktioniert AWS OpsWorks CM mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf AWS OpsWorks CM verwenden, sollten Sie wissen, welche IAM-Funktionen für die Verwendung mit CM verfügbar sind. AWS OpsWorks Einen allgemeinen Überblick darüber, wie AWS OpsWorks CM und andere AWS Dienste mit IAM funktionieren, finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Themen

- AWS OpsWorks Identitätsbasierte CM-Richtlinien
- AWS OpsWorks CM- und ressourcenbasierte Richtlinien
- Autorisierung basierend auf CM-Tags AWS OpsWorks
- AWS OpsWorks CM IAM-Rollen

AWS OpsWorks Identitätsbasierte CM-Richtlinien

Mit identitätsbasierten IAM-Richtlinien können Sie zulässige oder verweigerte Aktionen und Ressourcen sowie die Bedingungen angeben, unter denen Aktionen zugelassen oder verweigert werden. AWS OpsWorks CM unterstützt bestimmte Aktionen, Ressourcen und Bedingungsschlüssel. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

In AWS OpsWorks CM können Sie eine benutzerdefinierte Richtlinienerklärung an einen Benutzer, eine Rolle oder eine Gruppe anhängen.

Aktionen

Das Element `Action` einer identitätsbasierten IAM-Richtlinie beschreibt die spezifischen Aktionen, die von der Richtlinie zugelassen oder abgelehnt werden. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Die Aktion wird in einer Richtlinie verwendet, um Berechtigungen zur Durchführung der zugehörigen Aktion zu gewähren.

Richtlinienaktionen in AWS OpsWorks CM verwenden das folgende Präfix vor der Aktion:`opsworks-cm:`. Um jemandem beispielsweise die Berechtigung zum Erstellen eines AWS OpsWorks CM-Servers mithilfe einer API-Operation zu erteilen, fügen Sie die `opsworks-cm:CreateServer`-Aktion in seine Richtlinie ein. Richtlinienerklärungen müssen `Action` entweder ein `NotAction` Oder-Element enthalten. AWS OpsWorks CM definiert eigene Aktionen, die Aufgaben beschreiben, die Sie mit diesem Dienst ausführen können.

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie wie folgt durch Kommata:

```
"Action": [
      "opsworks-cm:action1",
      "opsworks-cm:action2"
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "opsworks-cm:Describe*"
```

Wenn Sie Platzhalter verwenden, um mehrere Aktionen in einer Richtlinienanweisung zuzulassen, achten Sie darauf, dass Sie diese Aktionen nur für autorisierte Services oder Benutzer zulassen.

Eine Liste der AWS OpsWorks CM-Aktionen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS OpsWorks im IAM-Benutzerhandbuch.

Ressourcen

Das Element `Resource` gibt die Objekte an, auf die die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource`- oder ein `NotResource`-Element enthalten. Sie geben eine Ressource unter Verwendung eines ARN oder eines Platzhalters (*) an, um anzugeben, dass die Anweisung für alle Ressourcen gilt.

Sie können die Amazon-Ressourcennummer (ARN) eines AWS OpsWorks CM-Servers oder -Backups abrufen, indem Sie die DescribeServersoder DescribeBackupsAPI-Operationen ausführen und Richtlinien auf Ressourcenebene für diese Ressourcen festlegen.

Eine AWS OpsWorks CM-Serverressource hat einen ARN im folgenden Format:

```
arn:aws:opsworks-cm:{Region}:${Account}:server/${ServerName}/${UniqueId}
```

Eine AWS OpsWorks CM-Backup-Ressource hat einen ARN im folgenden Format:

```
arn:aws:opsworks-cm:{Region}:${Account}:backup/${ServerName}-{Date-and-Time-Stamp-of-Backup}
```

Weitere Informationen zum Format von ARNs finden Sie unter Amazon Resource Names (ARNs) und AWS Service Namespaces.

Um beispielsweise den `test-chef-automate` Chef Automate-Server in Ihrer Anweisung anzugeben, verwenden Sie den folgenden ARN:

```
"Resource": "arn:aws:opsworks-cm:us-west-2:123456789012:server/test-chef-automate/
EXAMPLE-d1a2bEXAMPLE"
```

Um alle AWS OpsWorks CM-Server anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie den Platzhalter (*):

```
"Resource": "arn:aws:opsworks-cm:us-west-2:123456789012:server/*"
```

Im folgenden Beispiel wird ein AWS OpsWorks CM-Server-Backup als Ressource angegeben:

```
"Resource": "arn:aws:opsworks-cm:us-west-2:123456789012:backup/test-chef-automate-
server-2018-05-20T19:06:12.399Z"
```

Einige AWS OpsWorks CM-Aktionen, z. B. solche zum Erstellen von Ressourcen, können nicht für eine bestimmte Ressource ausgeführt werden. In diesen Fällen müssen Sie den Platzhalter (*) verwenden.

```
"Resource": "*"
```

Viele -API-Aktionen umfassen mehrere Ressourcen. Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie die ARNs durch Kommata voneinander.

```
"Resource": [
      "resource1",
      "resource2"
```

Eine Liste der AWS OpsWorks CM-Ressourcentypen und ihrer ARNs finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS OpsWorks CM im IAM-Benutzerhandbuch. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS OpsWorks CM im IAM-Benutzerhandbuch.

Bedingungsschlüssel

AWS OpsWorks CM verfügt nicht über servicespezifische Kontextschlüssel, die `Condition` in Richtlinienerklärungen verwendet werden können. Eine Liste der globalen Kontextschlüssel, die für alle Dienste verfügbar sind, finden Sie unter Kontextschlüssel für AWS globale Bedingungen in der IAM-Richtlinienreferenz. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungs-Operatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen `OR` Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Sie können einem Benutzer beispielsweise nur dann Zugriff auf eine Ressource gewähren, wenn sie mit dem Namen des Benutzers gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags (Markierungen) im IAM-Benutzerhandbuch.

Beispiele

Beispiele für identitätsbasierte AWS OpsWorks CM-Richtlinien finden Sie unter. AWS OpsWorks Beispiele für identitätsbasierte CM-Richtlinien

AWS OpsWorks CM- und ressourcenbasierte Richtlinien

AWS OpsWorks CM unterstützt keine ressourcenbasierten Richtlinien.

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die angeben, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für eine Ressource ausführen kann.

Autorisierung basierend auf CM-Tags AWS OpsWorks

Sie können Tags an AWS OpsWorks CM-Ressourcen anhängen oder Tags in einer Anfrage an AWS OpsWorks CM übergeben. Um den Zugriff auf der Basis von Tags zu steuern, stellen

Sie im Bedingungselement einer Richtlinie unter Verwendung der Bedingungsschlüssel `aws:RequestTag/key-name` oder `aws:TagKeys` Informationen zu Tags bereit. Weitere Informationen zum Taggen von AWS OpsWorks CM-Ressourcen finden Sie unter Mit Tags auf AWS OpsWorks for Chef Automate Ressourcen arbeiten oder Mit Tags auf AWS OpsWorks for Puppet Enterprise Ressourcen arbeiten in diesem Handbuch.

AWS OpsWorks CM IAM-Rollen

Eine IAM-Rolle ist eine Entität in Ihrem AWS Konto, die über bestimmte Berechtigungen verfügt.

AWS OpsWorks CM verwendet zwei Rollen:

- Eine Servicerolle, die dem AWS OpsWorks CM-Dienst Berechtigungen für die Arbeit innerhalb eines AWS Benutzerkontos gewährt. Wenn Sie die von OpsWorks CM bereitgestellte Standarddienstrolle verwenden, lautet der Name dieser Rolle`aws-opsworks-cm-service-role`.
- Eine Instanzprofilrolle, mit der der AWS OpsWorks CM-Dienst die OpsWorks CM-API aufrufen kann. Diese Rolle gewährt Zugriff auf Amazon S3 und AWS CloudFormation die Erstellung des Servers und des S3-Buckets für Backups. Wenn Sie das von OpsWorks CM bereitgestellte Standard-Instance-Profil verwenden, lautet der Name dieser Instance-Profilrolle`aws-opsworks-cm-ec2-role`.

AWS OpsWorks CM verwendet keine dienstbezogenen Rollen.

Verwenden temporärer Anmeldeinformationen mit AWS OpsWorks CM

AWS OpsWorks CM unterstützt die Verwendung temporärer Anmeldeinformationen und erbt diese Funktion von. AWS Security Token Service

Sie können temporäre Anmeldeinformationen verwenden, um sich über einen Verbund anzumelden, eine IAM-Rolle anzunehmen oder eine kontenübergreifende Rolle anzunehmen. Sie erhalten temporäre Sicherheitsanmeldedaten, indem Sie AWS STS API-Operationen wie AssumeRoleoder GetFederationTokenaufrufen.

Serviceverknüpfte Rollen

AWS OpsWorks CM verwendet keine dienstbezogenen Rollen.

Mit dienstverknüpften Rollen können AWS Dienste auf Ressourcen in anderen Diensten zugreifen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem IAM-

Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.

Servicerollen

Dieses Feature ermöglicht einem Service das Annehmen einer Servicerolle in Ihrem Namen. Diese Rolle gewährt dem Service Zugriff auf Ressourcen in anderen Diensten, um eine Aktion in Ihrem Namen auszuführen. Servicerollen werden in Ihrem IAM-Konto angezeigt und gehören zum Konto. Dies bedeutet, dass ein IAM-Administrator die Berechtigungen für diese Rolle ändern kann. Dies kann jedoch die Funktionalität des Dienstes beeinträchtigen.

AWS OpsWorks CM verwendet zwei Rollen:

- Eine Servicerolle, die dem AWS OpsWorks CM-Dienst Berechtigungen für die Arbeit innerhalb eines AWS Benutzerkontos gewährt. Wenn Sie die von OpsWorks CM bereitgestellte Standarddienstrolle verwenden, lautet der Name dieser Rolle`aws-opsworks-cm-service-role`.
- Eine Instanzprofilrolle, mit der der AWS OpsWorks CM-Dienst die OpsWorks CM-API aufrufen kann. Diese Rolle gewährt Zugriff auf Amazon S3 und AWS CloudFormation die Erstellung des Servers und des S3-Buckets für Backups. Wenn Sie das von OpsWorks CM bereitgestellte Standard-Instance-Profil verwenden, lautet der Name dieser Instance-Profilrolle`aws-opsworks-cm-ec2-role`.

Auswahl einer IAM-Rolle in CM AWS OpsWorks

Wenn Sie einen Server in AWS OpsWorks CM erstellen, müssen Sie eine Rolle auswählen, damit AWS OpsWorks CM in Ihrem Namen auf Amazon EC2 zugreifen kann. Wenn Sie bereits eine Servicerolle erstellt haben, stellt Ihnen AWS OpsWorks CM eine Liste von Rollen zur Auswahl zur Verfügung. OpsWorks CM kann die Rolle für Sie erstellen, wenn Sie keine angeben. Es ist wichtig, eine Rolle zu wählen, die Zugriff zum Starten und Stoppen von Amazon-EC2-Instances ermöglicht. Weitere Informationen finden Sie unter Erstellen eines Chef Automate-Servers oder Erstellen eines Puppet Enterprise-Masters.

AWS OpsWorks Beispiele für identitätsbasierte CM-Richtlinien

Standardmäßig sind Benutzer oder Rollen nicht berechtigt, AWS OpsWorks CM-Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mit der AWS Management Console AWS CLI, oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die IAM-

Identitäten die Erlaubnis gewähren, bestimmte API-Operationen auf den angegebenen Ressourcen auszuführen, die sie benötigen. Der Administrator muss diese Richtlinien anschließend den - Benutzern oder -Gruppen anfügen, die diese Berechtigungen benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

In AWS OpsWorks CM können Sie die `AWSOpsWorksCMServiceRole` Richtlinie einem Benutzer zuweisen, sodass der Benutzer Chef Automate- oder Puppet Enterprise-Server entweder mithilfe von oder erstellen und verwalten kann. AWS Management Console AWS CLI

Themen

- Bewährte Methoden für Richtlinien
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer
- AWS OpsWorks CM-Server anhand von Tags anzeigen

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien bestimmen, ob jemand OpsWorks CM-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum

Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.

- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung zum IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Konfigurieren eines MFA-geschützten API-Zugriffs im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die -Benutzern die Berechtigung zum Anzeigen der Inline-Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API oder. 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": "*"
        }
    ]
}
```

AWS OpsWorks CM-Server anhand von Tags anzeigen

Sie können Bedingungen in Ihrer identitätsbasierten Richtlinie verwenden, um den Zugriff auf AWS OpsWorks CM-Server und Backups anhand von Tags zu steuern. Dieses Beispiel zeigt, wie Sie eine Richtlinie erstellen könnten, die die Anzeige eines AWS OpsWorks CM-Servers ermöglicht. Die Berechtigung wird jedoch nur erteilt, wenn das AWS OpsWorks CM-Server-Tag den Wert des Benutzernamens dieses Benutzers `Owner` hat. Diese Richtlinie gewährt auch die Berechtigungen, die für die Ausführung dieser Aktion auf der Konsole erforderlich sind.

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

Sie können diese Richtlinie den -Benutzern in Ihrem Konto zuweisen. Wenn ein benannter Benutzer `richard-roe` versucht, einen AWS OpsWorks CM-Server aufzurufen, muss der Server mit `0wner=richard-roe` oder gekennzeichnet werden`owner=richard-roe`. Andernfalls wird der Zugriff abgelehnt. Der Tag-Schlüssel `0wner` der Bedingung stimmt sowohl mit `0wner` als auch mit `owner` überein, da die Namen von Bedingungsschlüsseln nicht zwischen Groß- und Kleinschreibung unterscheiden. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.

Problembehandlung bei AWS OpsWorks CM Identity and Access

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit IAM auftreten können. Informationen zur Problembehandlung speziell für AWS OpsWorks CM finden Sie unter Problembehebung AWS OpsWorks for Chef Automate undProblembehandlung OpsWorks für Puppet Enterprise.

Themen

- Ich bin nicht berechtigt, eine Aktion in AWS OpsWorks CM durchzuführen
- Ich bin nicht berechtigt, iam auszuführen: PassRole
- Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine AWS OpsWorks CM-Ressourcen ermöglichen

Ich bin nicht berechtigt, eine Aktion in AWS OpsWorks CM durchzuführen

Wenn Ihnen AWS Management Console mitgeteilt wird, dass Sie nicht berechtigt sind, eine Aktion auszuführen, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Der folgende Beispielfehler tritt auf, wenn ein Benutzer `mateojackson` versucht, die Konsole zu verwenden, um Details zu einem AWS OpsWorks CM-Server anzuzeigen, aber nicht über die `opsworks-cm:DescribeServers` entsprechenden Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 opsworks-cm:DescribeServers on resource: test-chef-automate-server
```

In diesem Fall bittet Mateo seinen Administrator, Richtlinien zu aktualisieren, damit er mit der `opsworks-cm:DescribeServers`-Aktion auf die `test-chef-automate-server`-Ressource zugreifen kann.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zur Ausführung der `iam:PassRole` Aktion autorisiert sind, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt. Bitten Sie diese Person, Ihre Richtlinien zu aktualisieren, damit Sie eine Rolle an OpsWorks CM übergeben können.

Bei einigen AWS Diensten können Sie eine bestehende Rolle an diesen Dienst übergeben, anstatt eine neue Servicerolle oder eine dienstbezogene Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in OpsWorks CM auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Service-Rolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall bittet Mary ihren Administrator um die Aktualisierung ihrer Richtlinien, um die Aktion `iam:PassRole` ausführen zu können.

Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine AWS OpsWorks CM-Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- AWS OpsWorks CM unterstützt die Gewährung von Zugriff auf Benutzer mit mehr als einem Konto, um einen AWS OpsWorks CM-Server zu verwalten.
- Informationen dazu, wie Sie den Zugriff auf Ihre Ressourcen mit Ihren AWS Konten gewähren können, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen AWS Konto, das Sie besitzen.
- Informationen dazu, wie Sie AWS Konten von Drittanbietern Zugriff auf Ihre Ressourcen gewähren, finden Sie im IAM-Benutzerhandbuch unter Zugriff auf AWS Konten, die Dritten gehören.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

AWSVon verwaltete Richtlinien fürAWS OpsWorks-Konfigurationsmanagement

Um Benutzern, Gruppen und Rollen Berechtigungen hinzuzufügen, ist es einfacher, von AWS verwaltete Richtlinien zu verwenden, als selbst Richtlinien zu schreiben. Es erfordert Zeit und Fachwissen, um von Kunden verwaltete IAM-Richtlinien zu erstellen, die Ihrem Team nur die benötigten Berechtigungen bieten. Um schnell loszulegen, können Sie unsere von AWS verwalteten Richtlinien verwenden. Diese Richtlinien decken häufige Anwendungsfälle ab und sind in Ihrem AWS-Konto verfügbar. Weitere Informationen zu verwalteten AWS-Richtlinien finden Sie unter Verwaltete AWS-Richtlinien im IAM-Leitfaden.

AWS-Services pflegen und Aktualisieren von verwalteten AWS-Richtlinien. Die Berechtigungen in von AWS verwalteten Richtlinien können nicht geändert werden. Services fügen einer von AWS verwalteten Richtlinien gelegentlich zusätzliche Berechtigungen hinzu, um neue Funktionen zu unterstützen. Diese Art von Update betrifft alle Identitäten (Benutzer, Gruppen und Rollen), an welche die Richtlinie angehängt ist. Services aktualisieren eine von AWS verwaltete Richtlinie am ehesten, wenn eine neue Funktion gestartet wird oder neue Vorgänge verfügbar werden. Services entfernen keine Berechtigungen aus einer von AWS verwalteten Richtlinie, so dass Richtlinien-Aktualisierungen Ihre vorhandenen Berechtigungen nicht beeinträchtigen.

Darüber hinaus unterstützt AWS verwaltete Richtlinien für Auftragsfunktionen, die mehrere Services umfassen. Zum Beispiel, dasReadOnlyZugriff auf AWSDie verwaltete Richtlinie bietet schreibgeschützten Zugriff auf alleAWSDienstleistungen und -Ressourcen. Wenn ein Service eine neue Funktion einführt, fügt AWS schreibgeschützte Berechtigungen für neue Operationen und Ressourcen hinzu. Eine Liste und Beschreibungen der Richtlinien für Auftragsfunktionen finden Sie in Verwaltete AWS-Richtlinien für Auftragsfunktionen im IAM-Leitfaden.

Von AWS verwaltete Richtlinie:**`AWSOpsWorksCMServiceRole`**

Sie können `AWSOpsWorksCMServiceRole` an Ihre IAM-Entitäten anhängen. OpsWorksCM fügt diese Richtlinie auch an eine Servicerolle an, dieOpsWorksCM um Aktionen in Ihrem Namen auszuführen.

Diese Richtlinie gewährt*`administrativ`*Berechtigungen, die zulassenOpsWorksCM—Administratoren zum Erstellen, Verwalten und LöschenOpsWorksCM-Server und Backups.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `opsworks-cm`— Ermöglicht es Prinzipalen, vorhandene Server zu löschen und Wartungsläufe zu starten.
- `acm`— Auf diese Weise können Prinzipale Zertifikate löschen oder importierenAWS Certificate Managermit denen Benutzer eine Verbindung zu einemOpsWorksCM-Server.
- `cloudformation`- ZulässtOpsWorksCM zum Erstellen und VerwaltenAWS CloudFormationStapel beim Erstellen, Aktualisieren oder LöschenOpsWorksCM-Server.
- `ec2`- ZulässtOpsWorksCM zum Starten, Bereitstellen, Aktualisieren und Beenden von Amazon Elastic Compute Cloud-Instanzen, wenn Prinzipale erstellen, aktualisieren oder löschenOpsWorksCM-Server.

- `iam`- ZulässtOpsWorksCM zum Erstellen von Servicerollen, die zum Erstellen und Verwalten erforderlich sindOpsWorksCM-Server.
- `tag`— Ermöglicht es Prinzipalen, Tags anzuwenden und zu entfernenOpsWorksCM-Ressourcen, einschließlich Server und Backups.
- `s3`- ZulässtOpsWorksCM zum Erstellen von Amazon S3 S3-Buckets zum Speichern von Server-Backups, zum Verwalten von Objekten in S3-Buckets auf Hauptanforderung (z. B. zum Löschen einer Sicherung) und zum Löschen von Buckets.
- `secretsmanager`- ZulässtOpsWorksCM zum Erstellen und Verwalten von Secrets Manager Manager-Geheimnissen und zum Anwenden oder Entfernen von Tags aus Secrets.
- `ssm`- ZulässtOpsWorksCM zum Verwenden von Systems Manager Run Command für Instances, dieOpsWorksCM-Server.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::aws-opsworks-cm-*"
            ],
            "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:::aws-opsworks-cm-*"
            ],
            "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/*"
            ]
        }
    ]
}
```

Von AWS verwaltete Richtlinie:**AWSOpsWorksCMInstanceProfileRole**

Sie können `AWSOpsWorksCMInstanceProfileRole` an Ihre IAM-Entitäten anhängen. OpsWorksCM fügt diese Richtlinie auch an eine Servicerolle an, dieOpsWorksCM um Aktionen in Ihrem Namen auszuführen.

Diese Richtlinie gewährt*administrativ*Berechtigungen, die Amazon EC2 EC2-Instances zulassen, die alsOpsWorksCM-Server, von denen Informationen abgerufen werdenAWS CloudFormationundAWS Secrets Managerund speichern Sie Server-Backups in Amazon S3 S3-Buckets.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `acm`- ZulässtOpsWorksCM-Server EC2-Instanzen, von denen Zertifikate abgerufen werdenAWS Certificate Managermit denen Benutzer eine Verbindung zu einemOpsWorksCM-Server.
- `cloudformation`- ZulässtOpsWorksCM-Server EC2-Instances zum Abrufen von InformationenAWS CloudFormationstapelt während des Erstellungs- oder Aktualisierungsprozesses der Instanz und sendet Signale anAWS CloudFormationüber seinen Status.

- `s3`- ZulässtOpsWorksCM-Server EC2-Instanzen zum Hochladen und Speichern von Serversicherungen in S3-Buckets, Stoppen oder Roll-Back-Uploads bei Bedarf und Löschen von Backups aus S3-Buckets.
- `secretsmanager`- ZulässtOpsWorksCM-Server EC2-Instanzen, um die Werte vonOpsWorksCM —bezogene Secrets Manager

```
{
    "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:::aws-opsworks-cm-*",
            "Effect": "Allow"
        },
        {
            "Action": "acm:GetCertificate",
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "arn:aws:secretsmanager:*:*:opsworks-cm!aws-opsworks-cm-
secrets-*",
```

```
            "Effect": "Allow"
        }
    ]
}
```

OpsWorksCM aktualisiert aufAWSVerwaltete Richtlinien

Anzeigen von Details zu Aktualisierungen zuAWSVon verwaltete Richtlinien fürOpsWorksCM seit dieser Service mit der Verfolgung dieser Änderungen begonnen hat. Um automatische Warnungen über Änderungen an dieser Seite erhalten, abonnieren Sie den RSS-Feed aufOpsWorksCM Dokumentverlaufangezeigten.

Änderung	Beschreibung	Datum
AWSOpsWorksCMInstanceProfileRole- Aktualisierung der verwalteten Richtlinien	OpsWorksCM hat die verwaltete Richtlinie aktualisiert, die die EC2-Instanzen zulässt, die alsOpsWorksCM-Server zum InformationsaustauschCloudFormationund Secrets Manager und verwalten Sie Backups. Die Änderung fügt hinzu`opsworks-cm!` zum Ressourcennamen für Secrets Manager Manager-Geheimnisse, damitOpsWorksCM darf die Geheimnisse besitzen.	23. April 2021
AWSOpsWorksCMServiceRole- Aktualisierung der verwalteten Richtlinien	OpsWorksCM hat die verwaltete Richtlinie aktualisiert, die es zulässtOpsWorksCM—Administratoren zum Erstellen, Verwalten und LöschenOpsWorksCM-Server und Backups. Die Änderung fügt hinzu`opsworks-`	23. April 2021

Änderung	Beschreibung	Datum
	cm! zum Ressourcennamen für Secrets Manager Manager-Geheimnisse, damitOpsW orksCM darf die Geheimnisse besitzen.	
OpsWorksCM hat mit Verfolgung von Änderungen	OpsWorksCM hat mit Verfolgung von Änderunge n für seineAWSVerwaltete Richtlinien.	23. April 2021

Dienstübergreifende verwirrter Stellvertreter-Prävention inAWS OpsWorks CM

Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine Entität, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine Entität mit größeren Rechten zwingen kann, die Aktion auszuführen. In AWS kann der dienstübergreifende Identitätswechsel zu Confused-Deputy-Problem führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen die Verwendung der globalen Bedingungskontext-Schlüssel `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Richtlinien, um die Berechtigungen, die AWS OpsWorks CM einem anderen Service erteilt, auf eine bestimmte Ressource zu beschränken. Wenn der `aws:SourceArn`-Wert die Konto-ID nicht enthält, z. B. einen Amazon-S3-Bucket-ARN, müssen Sie beide globale Bedingungskontextschlüssel verwenden, um Berechtigungen einzuschränken. Wenn Sie beide globale Bedingungskontextschlüssel verwenden und der `aws:SourceArn`-Wert die Konto-ID enthält, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in der gleichen Richtlinienanweisung verwendet wird. Verwenden Sie `aws:SourceArn`, wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten. Verwenden Sie `aws:SourceAccount`, wenn Sie zulassen möchten, dass Ressourcen in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft werden.

Der Wert von`aws:SourceArn`muss der ARN eines OpsWorksCM Chef- oder Puppet-Server.

Der effektivste Weg, um sich vor dem Confuse-Deputy-Problem zu schützen, ist die Verwendung des`aws:SourceArn`globaler Condition-Kontextschlüssel mit dem vollständigen ARN derAWS OpsWorks CMserver. Wenn Sie den vollständigen ARN nicht kennen oder wenn Sie mehrere Server-ARNs angeben, verwenden Sie die`aws:SourceArn`globaler Kontextbedingungsschlüssel mit Platzhaltern (*) für die unbekannten Teile des ARN. Zum Beispiel `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Der folgende Abschnitt zeigt, wie Sie die`aws:SourceArn`und`aws:SourceAccount`Globale Bedingungskontextschlüssel inAWS OpsWorks CMum das verwirrte Stellvertreter-Problem zu verhindern.

Verhindern Sie verwirrte stellvertretende HeldentatenAWS OpsWorks CM

In diesem Abschnitt wird beschrieben, wie Sie dazu beitragen können, verwirrte Stellvertreter-Exploits inAWS OpsWorks CMund enthält Beispiele für Berechtigungsrichtlinien, die Sie an die IAM-Rolle anhängen können, auf die Sie zugreifenAWS OpsWorks CMaus. Als bewährte Sicherheitsmethode empfehlen wir, die`aws:SourceArn`und`aws:SourceAccount`Bedingungsschlüssel für die Vertrauensbeziehungen, die Ihre IAM-Rolle mit anderen Diensten hat. Die Vertrauensbeziehungen erlaubenAWS OpsWorks CMum eine Rolle zu übernehmen, um Aktionen in anderen Diensten auszuführen, die zum Erstellen oder Verwalten IhrerAWS OpsWorks CM-Server.

So bearbeiten Sie Vertrauensbeziehungen**`aws:SourceArn`**und**`aws:SourceAccount`**-Bedingungsschlüssel

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. In derSuchesuchen Sie nach der Rolle, die Sie für den Zugriff verwendenAWS OpsWorks CMaus. DieAWSverwaltete Rolle ist`aws-opsworks-cm-service-role`aus.
4. Auf derÜbersicht-Seite für die Rolle, wählen Sie dieVertrauensstellungen-Registerkarte.
5. Wählen Sie auf der Registerkarte Trust relationships (Vertrauensstellungen) die Option Edit trust relationship (Vertrauensstellung bearbeiten).
6. In derRichtliniendokument, füge mindestens eine der`aws:SourceArn`oder`aws:SourceAccount`Bedingungsschlüssel für die Richtlinie. Verwenden von`aws:SourceArn`um die Vertrauensbeziehung zwischen Cross-Services einzuschränken (wieAWS Certificate Managerund Amazon EC2) undAWS OpsWorks CMauf

bestimmteAWS OpsWorks CM-Server, was restriktiver ist. Add`aws:SourceAccount`um das Vertrauensverhältnis zwischen Cross-Services einzuschränken undAWS OpsWorks CMauf Server in einem bestimmten Konto, was weniger restriktiv ist. Im Folgenden wird ein Beispiel gezeigt. Beachten Sie, dass die Konto-IDs identisch sein müssen, wenn Sie beide Bedingungsschlüssel verwenden.

```
{
  "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. Wenn Sie alle Bedingungsschlüssel hinzugefügt haben, wählen SieAktualisieren der Vertrauensrichtlinieaus.

Im Folgenden finden Sie weitere Beispiele für Rollen, die den Zugriff aufAWS OpsWorks CM-Server unter Verwendung von`aws:SourceArn`und`aws:SourceAccount`aus.

Themen

- Beispiel: ZugriffAWS OpsWorks CM-Server in einer bestimmten Region
- Beispiel: Hinzufügen von mehr als einem Server-ARN zuaws:SourceArn

Beispiel: ZugriffAWS OpsWorks CM-Server in einer bestimmten Region

Die folgende Anweisung zur Rollenvertrauensbeziehung greift auf eineAWS OpsWorks CM-Server in der Region USA Ost (Ohio) (`us-east-2`) enthalten. Beachten Sie, dass die Region im ARN-Wert von`aws:SourceArn`, aber der Wert der Server-ID ist ein Platzhalter (*).

```
{
  "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/*"
        }
      }
    }
  ]
}
```

Beispiel: Hinzufügen von mehr als einem Server-ARN zu**`aws:SourceArn`**

Das folgende Beispiel beschränkt den Zugriff auf ein Array von zweiAWS OpsWorks CMServer in der Konto-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"
           ]
       }
      }
    }
  ]
}
```

Richtlinie für den Datenverkehr zwischen Netzwerken

AWS OpsWorks CM verwendet dieselben Übertragungssicherheitsprotokolle, die üblicherweise von AWS verwendet werden: HTTPS oder HTTP mit TLS-Verschlüsselung.

Protokollierung und Überwachung in AWS OpsWorks CM

AWS OpsWorksCM protokolliert alle API-Aktionen unter. CloudTrail Weitere Informationen finden Sie unter den folgenden Themen:

- Protokollierung OpsWorks von Puppet Enterprise API-Aufrufen mit AWS CloudTrail
- AWS OpsWorks for Chef Automate API-Aufrufe protokollieren mit AWS CloudTrail

Compliance-Validierung für AWS OpsWorks CM

AWS OpsWorks CM unterstützt die folgenden Compliance-Programme und -Vorschriften:

- Payment Card Industry (PCI)
- Health Insurance Portability and Accountability Act (HIPAA) von 1996
- AWS System and Organization Controls (SOC) 1, 2 und 3.
- Datenschutz-Grundverordnung (DSGVO)

Externe Auditoren bewerten im Rahmen verschiedener AWS-Compliance-Programme die Sicherheit und Compliance von AWS OpsWorks CM. Hierzu zählen unter anderem SOC, PCI, FedRAMP und HIPAA.

Eine Liste der AWS-Services, die in bestimmten Compliance-Programmen enthalten sind, finden Sie unter AWS-Services in Scope nach Compliance-Programm. Allgemeine Informationen finden Sie unter AWS-Compliance-Programme.

Die Auditberichte von Drittanbietern lassen sich mit AWS Artifact herunterladen. Weitere Informationen finden Sie unter Herunterladen von Berichten in AWS Artifact.

Ihre Verantwortung in Bezug auf die Compliance bei der Verwendung von AWS OpsWorks CM wird durch die Vertraulichkeit Ihrer Daten, die Compliance-Ziele Ihres Unternehmens und die geltenden Gesetze und Vorschriften bestimmt. AWS stellt die folgenden Ressourcen zur Unterstützung bei Compliance-Fragen bereit:

- Kurzanleitungen für Sicherheit und Compliance – In diesen Bereitstellungsleitfäden finden Sie wichtige Überlegungen zur Architektur sowie die einzelnen Schritte zur Bereitstellung von sicherheits- und Compliance-orientierten Basisumgebungen in AWS.
- Whitepaper zur Erstellung einer Architektur mit HIPAA-konformer Sicherheit und Compliance – In diesem Whitepaper wird beschrieben, wie Unternehmen mithilfe von AWS HIPAA-konforme Anwendungen erstellen können.
- AWS-Compliance-Ressourcen – Diese Arbeitsbücher und Leitfäden könnten für Ihre Branche und Ihren Standort interessant sein.
- AWS Config – Dieser AWS-Service bewertet, zu welchem Grad die Konfiguration Ihrer Ressourcen den internen Vorgehensweisen, Branchenrichtlinien und Vorschriften entspricht.
- AWS Security Hub – Dieser AWS-Service liefert einen umfassenden Überblick über den Sicherheitsstatus in AWS. So können Sie die Compliance mit den Sicherheitsstandards in der Branche und den bewährten Methoden abgleichen.

Ausfallsicherheit in AWS OpsWorks CM

AWS OpsWorks CM ermöglicht standardmäßig tägliche Sicherungen von Servern, wenn Sie einen Server erstellen. Backups sind verschlüsselt und werden in einem Amazon S3 S3-Bucket gespeichert. Standardmäßig ist dieser Bucket nur für das Konto zugänglich, über das er Server erstellt wurde. Sie können nach eigenem Ermessen Bucket-Zugriff für andere Benutzer hinzufügen oder regionsübergreifende Backups in Amazon S3 konfigurieren. Chef und Puppet unterstützen

regionsübergreifende Verschlüsselung, da beide Produkte den Datenverkehr zwischen Ihrem AWS OpsWorks CM-Server und verwalteten Knoten verschlüsseln.

AWS OpsWorks CM unterstützt keine Hochverfügbarkeits-Konfigurationen (HA).

Im Zentrum der globalen AWS Infrastruktur stehen die AWS-Regionen und Availability Zones (Verfügbarkeitszonen, AZs). AWS Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die mit Netzwerken mit geringer Latenz, hohem Durchsatz und hochredundanten Vernetzungen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zum Sichern und Wiederherstellen von Servern in AWS OpsWorks CM finden Sie unter:

- Einen OpsWorks for Puppet Enterprise Server sichern und wiederherstellen
- Einen AWS OpsWorks for Chef Automate Server sichern und wiederherstellen

Weitere Informationen über AWS-Regionen und -Availability Zones finden Sie unter AWS Globale Infrastruktur.

Infrastruktursicherheit in AWS OpsWorks CM

Als verwalteter Service ist AWS OpsWorks Configuration Management durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS-Sicherheitsservices und wie AWS die Infrastruktur schützt, finden Sie unter AWS-Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS-Umgebung anhand der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastrukturschutz im Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf OpsWorks CM zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

AWS OpsWorks CM unterstützt keine PrivateLink oder privaten VPC-Endpunkte.

AWS OpsWorks CM bietet keine Unterstützung für ressourcenbasierte Richtlinien. Weitere Informationen finden Sie im AWS Identity and Access ManagementBenutzerhandbuch unter AWSDienste, die mit IAM funktionieren.

Konfigurations- und Schwachstellenanalyse in AWS OpsWorks CM

AWS OpsWorks CM führt regelmäßige Kernel- und Sicherheitsupdates für das Betriebssystem durch, das auf dem AWS OpsWorks CM-Server ausgeführt wird. Benutzer können ein Zeitfenster für automatische Updates festlegen, das bis zu zwei Wochen ab dem aktuellen Datum gültig ist. AWS OpsWorks CM veröffentlicht automatische Updates der Nebenversionen von Chef und Puppet Enterprise. Weitere Informationen zum Konfigurieren von Updates für AWS OpsWorks for Chef Automate finden Sie unter Systemwartung (Chef) in diesem Handbuch. Weitere Informationen zur Konfiguration von Updates OpsWorks für Puppet Enterprise finden Sie unter Systemwartung (Puppet) in diesem Handbuch.

Bewährte Sicherheitsmethoden für AWS OpsWorksCM

AWS OpsWorks CM bietet wie alle anderen AWS-Services Sicherheitsfunktionen, die beim Entwickeln und Implementieren Ihrer eigenen Sicherheitsrichtlinien zu berücksichtigen sind. Die folgenden bewährten Methoden sind allgemeine Richtlinien und keine vollständige Sicherheitslösung. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Überlegungen und nicht als bindend ansehen.

- Sichern Sie Ihr Starter Kit und laden Sie die Anmeldeinformationen herunter. Wenn Sie einen neuen AWS OpsWorks CM-Server erstellen oder ein neues Starter Kit und Anmeldeinformationen von der AWS OpsWorks CM-Konsole herunterladen, speichern Sie diese Elemente an einem sicheren Speicherort, der mindestens einen Authentifizierungsfaktor erfordert. Die Anmeldeinformationen bieten Zugriff auf Administratorebene auf Ihren Server.
- Sichern Sie Ihren Konfigurationscode. Sichern Sie Ihren Chef- oder Puppet-Konfigurationscode (Rezeptbücher und Module) mithilfe der empfohlenen Protokolle für Ihre Quell-Repositorys. Sie

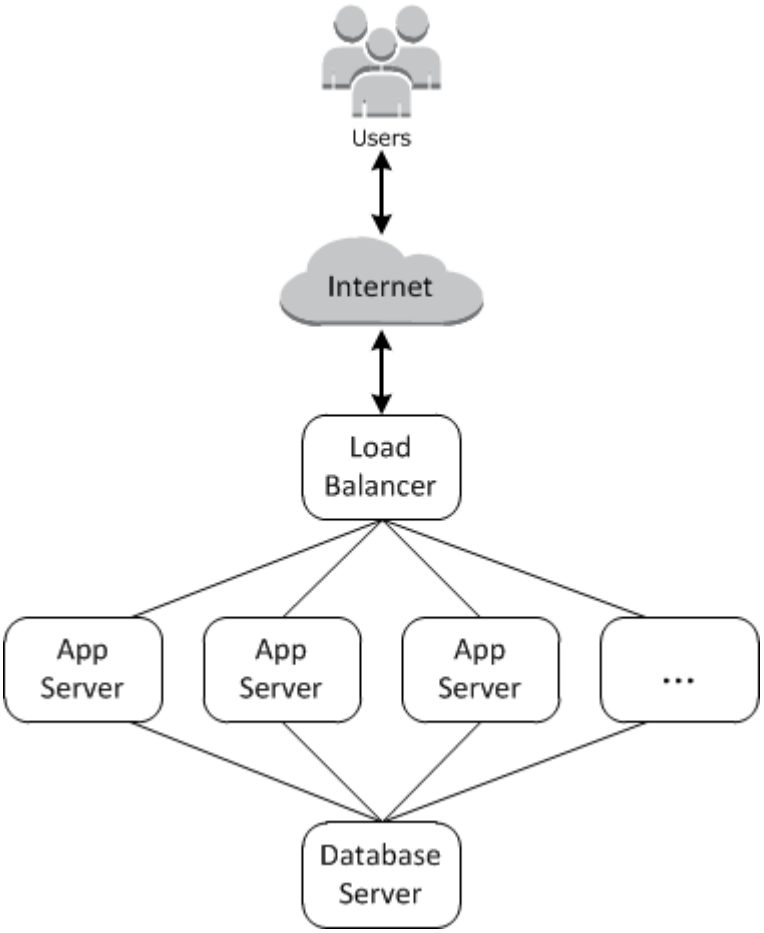
können beispielsweise die Berechtigungen auf Repositorys in AWS CodeCommit einschränken oder die Richtlinien auf der GitHub Website zur Sicherung von Repositorys befolgen. GitHub

- Verwenden Sie CA-signierte Zertifikate, um eine Verbindung mit Knoten herzustellen. Obwohl Sie selbstsignierte Zertifikate beim Registrieren oder Bootstrapping von Knoten auf dem AWS OpsWorks CM-Server verwenden können, verwenden Sie als bewährte Methode CA-signierte Zertifikate. Es wird empfohlen, ein Zertifikat zu verwenden, das von einer Zertifizierungsstelle (Certificate Authority, CA) signiert wurde.
- Geben Sie keine Anmeldeinformationen für die Chef- oder Puppet-Verwaltungskonsole für andere Benutzern frei. Ein Administrator sollte für jeden Benutzer der Chef- oder Puppet-Konsolen-Websites separate Benutzer erstellen.
  - Verwalten von Benutzern in Chef Automate
  - Verwalten von Benutzern in Puppet Enterprise
- Konfigurieren Sie automatische Sicherungen und Aktualisierungen der Systemwartung. Durch das Konfigurieren von automatischen Wartungsaktualisierungen auf dem AWS OpsWorks CM-Server wird sichergestellt, dass auf dem Server die neuesten sicherheitsrelevanten Betriebssystemupdates ausgeführt werden. Die Konfiguration automatischer Sicherungen erleichtert die Notfallwiederherstellung und verkürzte Wiederherstellungszeit im Falle eines Vorfalls oder eines Fehlers. Beschränken Sie den Zugriff auf den Amazon S3 S3-Bucket, in dem Ihre AWS OpsWorks CM-Server-Backups gespeichert sind. Gewähren Sie nicht Jedem Zugriff. Gewähren Sie anderen Benutzern nach Bedarf einzeln Lese- oder Schreibzugriff oder erstellen Sie eine Sicherheitsgruppe in IAM für diese Benutzer und weisen Sie der Sicherheitsgruppe Zugriff zu.
  - Systemwartung (Chef)
  - Systemwartung (Puppet)
  - Einen AWS OpsWorks for Chef Automate Server sichern und wiederherstellen
  - Einen OpsWorks for Puppet Enterprise Server sichern und wiederherstellen
  - Erstellen Ihres ersten delegierten Benutzers und Ihrer ersten delegierten Gruppe in IAM im AWS Identity and Access Management Benutzerhandbuch
  - Bewährte Sicherheitsmethoden für Amazon S3 im Amazon Simple Storage Service Developer Guide

AWS OpsWorks Stapel

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

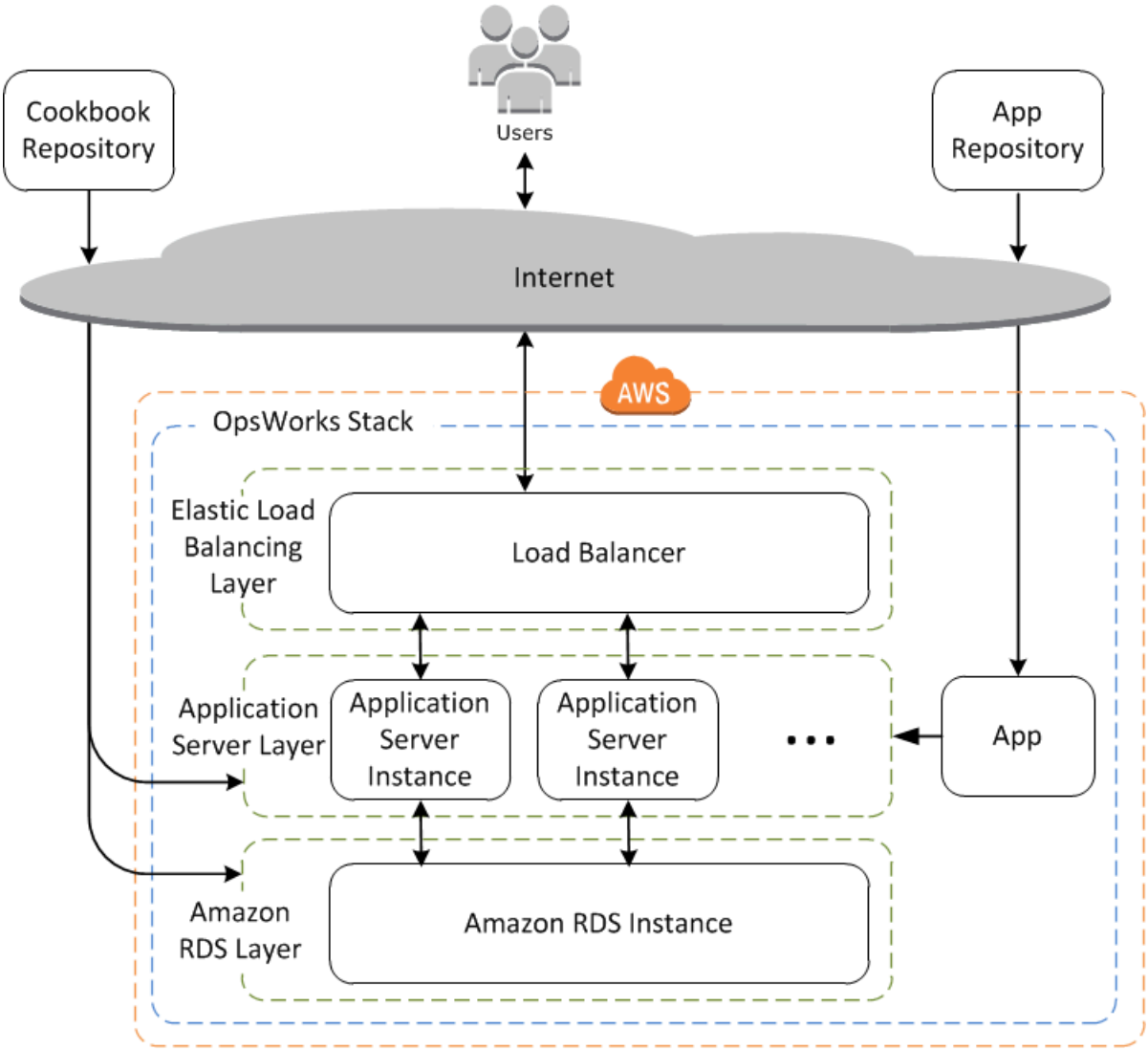
Cloud-basiertes Computing umfasst in der Regel Gruppen von AWS-Ressourcen, wie Amazon EC2 EC2-Instances und Amazon Relational Database Service (RDS) -Instances, die gemeinsam erstellt und verwaltet werden müssen. Beispielsweise sind für eine Webanwendung normalerweise Anwendungsserver, Datenbankserver, Load Balancer usw. erforderlich. Diese Gruppe von Instances wird üblicherweise als Stack bezeichnet. Ein einfacher Anwendungsserver-Stack sieht etwa wie folgt aus.



Zusätzlich zum Erstellen der Instances und dem Installieren der erforderlichen Pakete benötigen Sie in der Regel eine Möglichkeit zur Verteilung von Anwendungen auf die Anwendungsserver, zur Überwachung der Leistung, dem Verwalten der Sicherheit und Berechtigungen usw.

AWS OpsWorks Stacks bietet eine einfache und flexible Möglichkeit, Stacks und Anwendungen zu erstellen und zu verwalten.

So könnte ein einfacher Anwendungsserver-Stack mit AWS OpsWorks Stacks aussehen. Es besteht aus einer Gruppe von Anwendungsservern, die hinter einem Elastic Load Balancing Load Balancer laufen, mit einem Amazon RDS-Datenbankserver im Backend.



Dieser Stack ist zwar relativ einfach, zeigt aber alle wichtigen AWS OpsWorks Stacks-Funktionen. So wird es zusammengestellt.

Themen

- Stacks
- Ebenen
- Rezepte und Ereignisse LifeCycle
- Instances
- Apps
- Anpassen Ihres Stacks
- Ressourcenmanagement

Stacks

Der Stack ist die Kernkomponente von AWS OpsWorks Stacks. Es ist im Grunde ein Container für AWS-Ressourcen — Amazon EC2 EC2-Instances, Amazon RDS-Datenbank-Instances usw. —,

die einen gemeinsamen Zweck haben und logisch zusammen verwaltet werden sollten. Der Stack unterstützt Sie bei der Verwaltung dieser Ressourcen als Gruppe und definiert auch einige Standard-Konfigurationseinstellungen, wie beispielsweise das Betriebssystem der Instances und ihre AWS-Region. Wenn Sie einige Stack-Komponenten von der direkten Interaktion mit dem Benutzer isolieren möchten, können Sie den Stack in einer VPC ausführen.

Ebenen

Sie definieren die Stack-Komponenten, indem Sie einen oder mehrere Layer hinzufügen. Eine Ebene stellt eine Reihe von Amazon EC2 EC2-Instances dar, die einem bestimmten Zweck dienen, z. B. der Bereitstellung von Anwendungen oder dem Hosten eines Datenbankservers.

Sie können Layer anpassen oder erweitern, indem Sie die Standard-Konfigurationen von Paketen modifizieren, Chef-Rezepte zum Durchführen von Aufgaben, wie der Installation von zusätzlichen Paketen, hinzufügen und mehr.

Für alle Stacks umfasst AWS OpsWorks Stacks Service-Ebenen, die die folgenden AWS-Services darstellen.

- Amazon Relational Database Service
- Elastic Load Balancing
- Amazon Elastic Container Service

Layer geben Ihnen die vollständige Kontrolle darüber, welche Pakete installiert sind, wie sie konfiguriert sind, wie Anwendungen bereitgestellt werden und mehr.

Rezepte und Ereignisse LifeCycle

Die Layer sind von Chef-Rezepten abhängig zum Verarbeiten von Aufgaben, wie z. B. der Installation von Paketen auf Instances, der Bereitstellung von Anwendungen und dem Ausführen von Skripts usw. Eine der wichtigsten Funktionen von AWS OpsWorks Stacks ist eine Reihe von Lebenszyklusereignissen — Setup, Configure, Deploy, Undeploy und Shutdown —, durch die auf jeder Instanz automatisch ein bestimmter Satz von Rezepten zum richtigen Zeitpunkt ausgeführt wird.

Jeder Layer kann eine Reihe von Rezepten zu jedem Lebenszyklusereignis zugewiesen haben, die eine Vielzahl von Aufgaben für dieses Ereignis und Layer bearbeiten. Wenn beispielsweise eine

Instanz, die zu einer Webserver-Ebene gehört, den Startvorgang abgeschlossen hat, führt Stacks die folgenden Schritte aus. AWS OpsWorks

1. Das Ausführen von Einrichtungsrezepten des Layers, welche Aufgaben wie das Installieren und Konfigurieren eines Webservers durchführen könnten.
2. Das Ausführen der Bereitstellungsrezepte der einzelnen Layer, die der Instance die Layer-Anwendungen aus einem Repository bereitstellen und verwandte Aufgaben, wie z. B. das Neustarten des Dienstes, durchführen.
3. Das Ausführen der Konfigurierungsrezepte auf allen Instances in dem Stack, damit jede Instance die Konfiguration anpassen kann, um die neue Instance zu unterstützen.

   In einer Instance, die einen Load Balancer ausführt, könnte z. B. ein Konfigurationsrezept die Konfiguration des Load Balancers modifizieren, um so die neue Instance einzuschließen.

Wenn eine Instanz zu mehreren Ebenen gehört, führt AWS OpsWorks Stacks die Rezepte für jede Ebene aus, sodass Sie beispielsweise eine Instanz haben können, die einen PHP-Anwendungsserver und einen MySQL-Datenbankserver unterstützt.

Wenn Sie Rezepte implementiert haben, können Sie jedes Rezept der entsprechenden Ebene und dem entsprechenden Ereignis zuweisen, und AWS OpsWorks Stacks führt sie automatisch zum richtigen Zeitpunkt für Sie aus. Sie können Rezepte auch jederzeit manuell ausführen.

Instances

Eine Instance stellt eine einzelne Rechenressource dar, z. B. eine Amazon EC2 EC2-Instance. Sie definiert die grundlegende Konfiguration der Ressource, wie das Betriebssystem und die Größe. Andere Konfigurationseinstellungen, wie Elastic IP-Adressen oder Amazon EBS-Volumes, werden durch die Ebenen der Instance definiert. Die Layer-Rezepte schließen die Konfiguration ab, indem Sie Aufgaben wie die Installation, die Konfiguration von Paketen und die Bereitstellung von Apps erfüllen.

Sie können AWS OpsWorks Stacks verwenden, um Instances zu erstellen und sie einer Ebene hinzuzufügen. Wenn Sie die Instance starten, startet AWS OpsWorks Stacks eine Amazon EC2 EC2-Instance mit den von der Instance und ihrem Layer angegebenen Konfigurationseinstellungen. Nachdem der Start der Amazon EC2 EC2-Instance abgeschlossen ist, installiert AWS OpsWorks Stacks einen Agenten, der die Kommunikation zwischen der Instance und dem Service abwickelt und die entsprechenden Rezepte als Reaktion auf Lebenszyklusereignisse ausführt.

AWS OpsWorks Stacks unterstützt die folgenden Instance-Typen, die sich dadurch auszeichnen, wie sie gestartet und gestoppt werden.

- 24/7-Instances werden manuell gestartet und ausgeführt, bis Sie sie anhalten.
- Zeitbasierte Instances werden von AWS OpsWorks Stacks nach einem bestimmten Tages- und Wochenplan ausgeführt.

  Sie erlauben Ihrem Stack die Anzahl der Instances automatisch anzupassen, um vorhersehbare Nutzungsmuster zu unterstützen.
- Lastbasierte Instances werden automatisch von AWS OpsWorks Stacks gestartet und gestoppt, basierend auf bestimmten Lastmetriken, wie z. B. der CPU-Auslastung.

  Sie erlauben Ihrem Stack, die Anzahl der Instances automatisch anzupassen, um die Schwankungen des eingehenden Datenverkehrs zu unterstützen. Lastbasierte Instances sind nur für Linux-basierte Stacks verfügbar.

AWS OpsWorks Stacks unterstützt die automatische Heilung von Instanzen. Wenn ein Agent die Kommunikation mit dem Service beendet, stoppt AWS OpsWorks Stacks die Instanz automatisch und startet sie neu.

Sie können auch Linux-basierte Rechenressourcen in einen Stack integrieren, der außerhalb von Stacks erstellt wurde. AWS OpsWorks

- Amazon EC2 EC2-Instances, die Sie direkt mit der Amazon EC2 EC2-Konsole, CLI oder API erstellt haben.
- Lokale Instances, die auf Ihrer eigenen Hardware ausgeführt werden, einschließlich Instances auf virtuellen Maschinen.

Nachdem Sie eine dieser Instances registriert haben, wird sie zu einer AWS OpsWorks Stacks-Instance und Sie können sie auf die gleiche Weise verwalten wie Instances, die Sie mit Stacks erstellen. AWS OpsWorks

Apps

Sie speichern Anwendungen und zugehörige Dateien in einem Repository, z. B. einem Amazon S3 S3-Bucket. Jede Anwendung wird durch eine App repräsentiert, die den Anwendungstyp spezifiziert und die Informationen enthält, die erforderlich sind, um Ihre Anwendung aus dem Repository auf

Ihren Instances bereitzustellen, z. B. die Repository-URL und das Passwort. Wenn Sie eine App bereitstellen, löst AWS OpsWorks Stacks ein Deploy-Ereignis aus, das die Deploy-Rezepte auf den Instances des Stacks ausführt.

Sie können Anwendungen auf folgende Arten bereitstellen:

- Automatisch — Wenn Sie Instances starten, führt AWS OpsWorks Stacks automatisch die Deploy-Rezepte der Instanz aus.
- Manuell - Falls Sie eine neue Anwendung haben oder eine existierende aktualisieren möchten, können Sie die Bereitstellungsrezepte der Online-Instances manuell ausführen.

Normalerweise lassen Sie AWS OpsWorks Stacks die Deploy-Rezepte für den gesamten Stack ausführen, sodass die Instanzen der anderen Ebenen ihre Konfiguration entsprechend ändern können. Dennoch können Sie die Bereitstellung auf eine Teilmenge der Instances limitieren, z. B. wenn Sie eine neue Anwendung testen möchten, bevor sie für jede Anwendungsserver-Instance bereitgestellt wird.

Anpassen Ihres Stacks

Mit AWS OpsWorks Stacks haben Sie verschiedene Möglichkeiten, Layer an Ihre spezifischen Anforderungen anzupassen:

- Sie können ändern, wie AWS OpsWorks Stacks Pakete konfiguriert, indem Sie Attribute überschreiben, die die verschiedenen Konfigurationseinstellungen repräsentieren, oder indem Sie sogar die Vorlagen überschreiben, die zum Erstellen von Konfigurationsdateien verwendet wurden.
- Sie können einen vorhandenen Layer erweitern, indem Sie Ihre eigenen Rezepte zum Ausführen von Aufgaben, wie die Ausführung von Skripts oder die Installation und Konfiguration von Nichtstandard-Paketen, ausführen.

Alle Stacks können eine oder mehrere Layer enthalten, die am Anfang nur eine minimale Anzahl von Rezepten umfassen. Durch die Implementierung von Rezepten zum Bearbeiten von Aufgaben, wie die Installation von Paketen, die Bereitstellung von Anwendungen, können Sie dem Layer Funktionen hinzufügen. Sie verpacken Ihre benutzerdefinierten Rezepte und zugehörige Dateien in einem oder mehreren Kochbüchern und speichern die Kochbücher in einem Repository wie Amazon S3 oder Git.

Sie können Rezepte manuell ausführen, aber mit AWS OpsWorks Stacks können Sie den Prozess auch automatisieren, indem es eine Reihe von fünf Lebenszyklusereignissen unterstützt:

- Die Einrichtung erfolgt über eine neue Instance, nachdem sie erfolgreich gestartet wird.
- Die Konfiguration wird auf allen Instances der Stacks durchgeführt, wenn eine Instance in den Online-Status wechselt oder diesen verlässt.
- Die Bereitstellung findet statt, wenn Sie eine Anwendung bereitstellen.
- Die Bereitstellungsaufhebung tritt auf, wenn Sie eine Anwendung löschen.
- Das Herunterfahren wird ausgeführt, wenn Sie eine Instance anhalten.

Jeder Layer kann jedem Ereignis eine beliebige Anzahl an Rezepten zuordnen. Wenn ein Lebenszyklusereignis auf einer Layer-Instanz auftritt, führt AWS OpsWorks Stacks die zugehörigen Rezepte aus. Wenn beispielsweise ein Deploy-Ereignis auf einer App-Server-Instanz auftritt, führt AWS OpsWorks Stacks die Deploy-Rezepte des Layers aus, um die App herunterzuladen oder verwandte Aufgaben auszuführen.

Ressourcenmanagement

Sie können andere AWS-Ressourcen wie Elastic IP-Adressen in Ihren Stack integrieren. Sie können die AWS OpsWorks Stacks-Konsole oder -API verwenden, um Ressourcen bei einem Stack zu registrieren, registrierte Ressourcen an Instanzen anzuhängen oder von ihnen zu trennen und Ressourcen von einer Instanz zur anderen zu verschieben.

Sicherheit und Berechtigungen

AWS OpsWorks Stacks lässt sich in AWS Identity and Access Management (IAM) integrieren, um robuste Methoden zur Steuerung des Benutzerzugriffs auf AWS OpsWorks Stacks bereitzustellen, darunter die folgenden:

- Wie einzelne Benutzer mit jedem Stack interagieren können, z. B. ob sie Stack-Ressourcen wie Ebenen und Instances erstellen können oder ob sie SSH oder RDP verwenden können, um eine Verbindung zu den Amazon EC2 EC2-Instances eines Stacks herzustellen.
- Wie AWS OpsWorks Stacks in Ihrem Namen handeln kann, um mit AWS-Ressourcen wie Amazon EC2 EC2-Instances zu interagieren.
- Wie Apps, die auf AWS OpsWorks Stacks-Instances ausgeführt werden, auf AWS-Ressourcen wie Amazon S3 S3-Buckets zugreifen können.
- Wie RDP-Kennwörter und öffentliche SSH-Schlüssel der Benutzer zu verwalten sind und mit einer Instance verbunden werden.

Überwachung und Protokollierung

AWS OpsWorks Stacks bietet verschiedene Funktionen, mit denen Sie Ihren Stack überwachen und Probleme mit Ihrem Stack und allen Rezepten beheben können. Für alle Stacks:

- AWS OpsWorks Stacks bietet eine Reihe von benutzerdefinierten CloudWatch Metriken für Linux-Stacks, die der Einfachheit halber auf der Monitoring-Seite zusammengefasst sind.

  AWS OpsWorks Stacks unterstützt die CloudWatch Standardmetriken für Windows-Stacks. Sie können sie mit der CloudWatch Konsole überwachen.
- CloudTrail Protokolle, die API-Aufrufe von oder im Namen von AWS OpsWorks Stacks in Ihrem AWS-Konto aufzeichnen.
- Ein Ereignisprotokoll, das alle Ereignisse in Ihrem Stack auflistet.
- Chef-Protokolle, die aufführen, was sich für jedes Lebenszyklusereignis auf jeder Instance herausgestellt hat, z. B. welche Rezepte ausgeführt wurden und welche Fehler aufgetreten sind.

Linux-basierte Stacks können auch einen Ganglia-Master-Layer enthalten, mit dem Sie detaillierte Überwachungsdaten für die Instances in Ihrem Stack sammeln und anzeigen können.

CLI, SDK und AWS CloudFormation Vorlagen

Neben der Konsole unterstützt AWS OpsWorks Stacks auch eine Befehlszeilenschnittstelle (CLI) und SDKs für mehrere Sprachen, die für die Ausführung beliebiger Operationen verwendet werden können. Beachten Sie die folgenden Funktionen:

- Die AWS OpsWorks Stacks-CLI ist Teil der AWS-CLI und kann verwendet werden, um alle Operationen von der Befehlszeile aus auszuführen.

  Die AWS-CLI unterstützt mehrere AWS-Services und kann auf Windows-, Linux- oder OS X-Systemen installiert werden.
- AWS OpsWorks Stacks ist in AWS Tools for Windows enthalten PowerShell und kann verwendet werden, um beliebige Operationen von einer PowerShell Windows-Befehlszeile aus auszuführen.
- Das AWS OpsWorks Stacks SDK ist in den AWS-SDKs enthalten und kann von Anwendungen verwendet werden, die in folgenden Sprachen implementiert sind: Java, JavaScript(browserbasiert und Node.js), .NET, PHP, Python (Boto) oder Ruby.

Sie können auch AWS CloudFormation Vorlagen verwenden, um Stacks bereitzustellen. Einige Beispiele finden Sie unter OpsWorks AWS-Snippets.

AWS OpsWorks Stacks Häufig gestellte Fragen zum Lebensende

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren.

Themen

Wie werden Bestandskunden von diesem Ende ihrer Nutzungsdauer betroffen sein?

Bestandskunden sind bis zum 26. Mai 2024, dem Enddatum von, nicht betroffen. AWS OpsWorks Stacks Nach dem 26. Mai 2024 können Kunden die OpsWorks Konsole, API, CLI und CloudFormation Ressourcen nicht mehr verwenden.

Nimmt AWS OpsWorks Stacks er neue Kunden an?

Nein. AWS OpsWorks Stacks akzeptiert keine neuen Kunden mehr und nur Bestandskunden können derzeit neue Stacks erstellen.

Wohin sollte ich meine bestehenden Stacks migrieren?

Wir empfehlen AWS OpsWorks Stacks Kunden, ihre Workloads AWS Systems Manager dorthin zu migrieren, wo sie die folgenden Funktionen nutzen können:

- Moderne Chef-Versionen
- SSM-Agent
- Application Load Balancer
- Verbesserte Skalierungsfunktionen über Auto Scaling Scaling-Gruppen
- Möglichkeit, die gewünschten Hosteigenschaften mithilfe von EC2-Startvorlagen zu definieren
- Neuere Instance-Typen
- Neuere EBS-Volumetypen

Informationen zu Systems Manager finden Sie im AWS Systems Manager Benutzerhandbuch. Informationen zur Migration zu finden Sie AWS Systems Manager unter Migrieren Sie Ihre AWS OpsWorks Stacks Anwendungen zu AWS Systems Manager Application Manager

Wie kann ich meine bestehenden Amazon EC2 EC2-Instances nach dem Ende der Nutzungsdauer behalten?

Nach Ablauf des End-of-Life-Datums verbleiben Ihre Amazon EC2 EC2-Instances in Ihrem Konto, Sie können den OpsWorks Stacks-Service jedoch nicht mehr zur Steuerung und Verwaltung der Instances verwenden.

Sie können das Tool AWS OpsWorks Stacks Detach in Place verwenden, um Ihre OpsWorks Instances vom Stacks-Service zu trennen. OpsWorks Nach der Trennung können Sie Amazon EC2 oder einen anderen EC2-kompatiblen Ansatz verwenden AWS Systems Manager, um die Instances zu konfigurieren und zu verwalten. Weitere Informationen finden Sie unter Verwenden des Werkzeugs „An Ort und Stelle AWS OpsWorks Stacks lösen“.

Wirkt sich das Lebensende auf alle AWS-Regionen gleichzeitig aus?

Ja. Die OpsWorks Konsole, die API, die CLI und die CloudFormation Ressourcen werden am 26. Mai 2024 insgesamt AWS-Regionen gleichzeitig eingestellt. Eine Liste der verfügbaren AWS-Regionen Dienste AWS OpsWorks Stacks finden Sie unter Liste der AWS regionalen Dienste.

Für welches Maß an technischem Support steht es zur Verfügung AWS OpsWorks Stacks?

AWS wird bis zum Ende der Nutzungsdauer weiterhin AWS OpsWorks Stacks das gleiche Maß an Support bieten, das Kunden heute haben. Wenn Sie Fragen oder Bedenken haben, können Sie das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support kontaktieren.

Wird es neue Feature-Releases für geben? AWS OpsWorks Stacks

Nein. Da der Dienst das Ende seiner Nutzungsdauer erreicht, werden wir keine neuen Funktionen veröffentlichen. Wir werden jedoch weiterhin die Sicherheit verbessern und Amazon EC2 EC2-Instances wie erwartet bis zum Ende des Lebenszyklus verwalten.

Migrieren Sie Ihre AWS OpsWorks Stacks Anwendungen zu AWS Systems Manager Application Manager

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren.

Sie können Ihre AWS OpsWorks Stacks Anwendungen jetzt mithilfe eines Migrationsskripts zu Application Manager migrieren AWS Systems Manager, was eine Funktion von ist. Durch die Migration Ihrer Stacks-Anwendungen zu Systems Manager Application Manager können Sie AWS Funktionen nutzen, die in nicht verfügbar sind AWS OpsWorks Stacks, wie z. B. neue Amazon EC2 EC2-Instance-Typen wie Graviton, neue Amazon Elastic Block Store (EBS) -Volumes wie gp3, neue Betriebssysteme, Integrationen mit Auto Scaling Scaling-Gruppen und Anwendungs-Loadbalancer.

Mit dieser Version können Sie jetzt Operationen auf Ihren migrierten Instances mithilfe einer neuen Registerkarte „Instances“ überwachen und ausführen, die im Systems Manager Application Manager verfügbar ist. Sie können die Registerkarte „Instanzen“ verwenden, um mehrere AWS Instanzen an einem Ort anzuzeigen. Auf dieser Registerkarte können Sie Informationen zum Zustand der Instance einsehen und Probleme beheben. Weitere Informationen zur Arbeit mit dem Tab „Instanzen“ finden Sie im AWS Systems Manager Benutzerhandbuch unter Arbeiten mit Ihren Anwendungsinstanzen.

Themen

So funktioniert das Skript

AWS OpsWorks stellt ein Skript bereit, das Sie ausführen können, um Ihre AWS OpsWorks Stacks Anwendungen mithilfe einer CloudFormation Vorlage zu Systems Manager Application Manager zu migrieren. Das Skript ruft Informationen über eine vorhandene OpsWorks Ebene ab und stellt je nach Wert des `--provision-application` Parameters für das Skript entweder einen Klon Ihrer Anwendung bereit oder stellt eine CloudFormation Startvorlage bereit, mit der Sie Änderungen vornehmen können AWS CloudFormation.

Voraussetzungen

- Stellen Sie sicher, dass das installiert und konfiguriert AWS CLI ist. Weitere Informationen zur AWS CLI Installation von finden Sie unter Installation oder Aktualisierung der neuesten Version von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

  > **Note**
  >
  > Wenn Sie das nicht konfigurieren möchten AWS CLI, können Sie Befehle auch mit ausführen AWS CloudShell. Weitere Informationen zum Arbeiten mit CloudShell finden Sie AWS CloudShell im AWS CloudShell Benutzerhandbuch unter Arbeiten mit.

- Stellen Sie sicher, dass Python Version 3.6 oder neuer installiert ist oder mit dem Amazon Machine Image (AMI) geliefert wird.
- Stellen Sie sicher, dass Ihr Betriebssystem unterstützt wird. Sie können das Migrationsskript unter den folgenden Betriebssystemen herunterladen und ausführen.
  - Amazon Linux und Amazon Linux 2
  - Ubuntu 18.04 LTS, 20.04 LTS, 22.04 LTS

- RedHat Enterprise Linux 8
- Windows Server 2019, Windows 10 Unternehmen

> **Note**
>
> Windows Server 2022 wird nicht unterstützt.

Einschränkungen

Die neue OpsWorks Architektur unterscheidet sich von der Architektur für AWS OpsWorks Stacks. In diesem Abschnitt werden die bekannten Einschränkungen dieser Architektur beschrieben.

Folgendes wird von der neuen OpsWorks Architektur nicht unterstützt.

- Chef-Rezepte auf Windows- und CentOS-Instanzen ausführen
- Integrierte Chef 11-Ebenen und Berkshelf
- Chef-Attribute und Datentaschen
- Lokale Instances
- Aus EC2 importierte Instanzen
- Keine Unterstützung für die Installation einer vom Benutzer angegebenen Liste von Betriebssystempaketen
- Apps werden nicht unterstützt oder migriert

Folgendes wird mit Einschränkungen unterstützt.

- Das Migrationsskript klont die EBS-Volume-Informationen, schließt jedoch Bereitstellungspunkte und die tatsächlichen Daten, die in den Volumes enthalten sind, aus.
- Zeit- und lastbasierte skalierte Instances werden migriert, aber alle Skalierungsregeln, die mit diesen Instances verknüpft sind, werden nicht migriert. Sie können die Auto Scaling Scaling-Gruppe ändern, um ähnliche Ergebnisse zu erzielen.
- IAM-Entitäten, die auf der Seite „Berechtigungen“ des Stacks in der OpsWorks Konsole definiert sind, werden nicht erstellt oder generiert.
- Das Migrationsskript kann nur Single-Layer-Anwendungen in Systems Manager bereitstellen. Wenn Sie das Skript beispielsweise zweimal für zwei Ebenen im selben Stapel ausführen, erhalten Sie zwei verschiedene Anwendungen in Systems Manager.

Erste Schritte

Das Migrationsskript`stack_exporter.py`,, ist ein Python-Skript, das Sie lokal oder auf einer EC2-Instance ausführen können. Stellen Sie vor der Ausführung des Skripts sicher, dass alle Voraussetzungen erfüllt sind. Weitere Informationen zu den Voraussetzungen finden Sie unterVoraussetzungen.

Die Schritte in den folgenden Abschnitten zeigen Ihnen, wie Sie Ihre OpsWorks Stacks zu Systems Manager Application Manager migrieren.

Themen

Schritt 1: Bereiten Sie Ihre Umgebung für die Ausführung des Skripts vor

Bereiten Sie Ihre Umgebung vor, indem Sie die entsprechenden Befehle für Ihr Betriebssystem ausführen.

Themen

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
```

Führen Sie für Python-Version 3.6 auch Folgendes aus:

```
python3 -m pip install pipenv==2022.4.8
```

Führen Sie für Python-Version 3.7 und neuer auch Folgendes aus:

```
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
```

RedHat 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 Unternehmen

> **Note**
>
> Installieren Sie für Windows Server 2019 Python Version 3.6.1 oder neuer.

```
pip install pipenv
```

Falls Git noch nicht installiert ist, laden Sie Git herunter und installieren Sie es.

Wenn Sie Git als Kochbuchquelle verwenden, fügen Sie Ihren Git-Server zu einer `known_hosts` Datei hinzu, bevor Sie das Skript unter Windows ausführen. Sie können verwenden PowerShell , um die folgende Funktion zu erstellen.

```
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."
   }
}
```

Sie können dann Ihren Git-Server angeben (z. B. github.com, git-codecommit). *repository_region* .amazonaws.com), wenn Sie die Funktion ausführen.

```
add_to_known_hosts "myGitServer"
```

Schritt 2: Laden Sie das Migrationsskript herunter

Laden Sie die ZIP-Datei mit dem Migrationsskript und allen relevanten Dateien herunter, indem Sie den folgenden Befehl ausführen.

```
aws s3api get-object \
    --bucket export-opsworks-stacks-bucket-prod-us-east-1 \
    --key export_opsworks_stacks_script.zip export_opsworks_stacks_script.zip
```

Wenn Sie Linux verwenden, installieren Sie das Unzip-Hilfsprogramm mit den folgenden Befehlen.

```
sudo apt-get install unzip
sudo yum install unzip
```

Entpacken Sie die Dateien mit dem entsprechenden Befehl für Ihr Betriebssystem.

Verwenden Sie für Linux den folgenden Befehl.

```
unzip export_opsworks_stacks_script.zip
```

Verwenden Sie für Windows den `Expand-Archive` Befehl in PowerShell.

```
Expand-Archive -LiteralPath PathToZipFile -DestinationPath PathToDestination
```

Nach dem Entpacken der Datei sind die folgenden Verzeichnisse und Dateien verfügbar.

- README.md
- LIZENZ
- NOTICE
- requirements.txt
- Vorlagen/
  - OpsWorkscfnTemplate.yaml
  - MountEbsVolumes.YAML

- opsworks/
- Wolkenbildung/
- instanzen_tab/
- cfn_stack_deployer.py
- s3.py
- stack_exporter_context.py
- stack_exporter.py

Schritt 3: Richten Sie Ihre Umgebung für die Ausführung des Skripts ein

Richten Sie Ihre Umgebung für die Ausführung des Skripts ein, indem Sie den folgenden Befehl verwenden.

```
pipenv install -r requirements.txt
pipenv shell
```

> **Note**
>
> Derzeit kann das Skript nur einschichtige Anwendungen in Application Manager bereitstellen. Wenn Sie das Skript beispielsweise zweimal für zwei Ebenen im selben Stapel ausführen, erstellt das Skript zwei verschiedene Anwendungen in Application Manager.

Nachdem Sie Ihre Umgebung eingerichtet haben, überprüfen Sie die Skriptparameter. Sie können die verfügbaren Optionen für das Migrationsskript anzeigen, indem `python3 stack_exporter.py --help` Sie den Befehl ausführen.

Parameter	Beschreibung	Erforderlich	Typ	Standardwert
`--layer-id`	Exportiert eine CloudFormation Vorlage für diese OpsWorks Layer-ID.	Ja	Zeichenfolge	
`--region`	Die AWS Region für den OpsWorks Stapel. Wenn sich Ihre	Nein	Zeichenfolge	us-east-1

Parameter	Beschreibung	Erforderlich	Typ	Standardwert
	OpsWorks Stack-Region und Ihre API-Endpunktregion unterscheiden, verwenden Sie die Stack-Region. Dies ist dieselbe Region wie die anderen Ressourcen in Ihrem OpsWorks Stack (z. B. EC2-Instances und Subnetze).			
`--provision-application`	Standardmäßig stellt das Skript die mit der Vorlage exportierte Anwendung bereit. CloudFormation Übergeben Sie diesen Parameter mit dem Wert FALSE an das Skript, um die Bereitstellung der CloudFormation Vorlage zu überspringen.	Nein	Boolesch	`TRUE`

Parameter	Beschreibung	Erforderlich	Typ	Standardwert
`--launch-template`	Dieser Parameter definiert, ob eine vorhandene Startvorlage verwendet oder eine neue Startvorlage erstellt werden soll. Sie können eine neue Startvorlage erstellen, die die empfohlenen Instanzeigenschaften verwendet oder die Instanzeigenschaften verwendet, die einer Online-Instance entsprechen. Gültige Werte sind: • `RECOMMENDED` - Verwendet Instance-Eigenschaften aus dem neuesten AMI für das Betriebssystem des OpsWorks Stacks und eine c5.large-Instance-Größe. • `MATCH_LAST_INSTANCE` - Verwendet die neuesten verfügbaren Online-Instance-Eigenschaften. • *`LaunchTemplateID`* / [*`LaunchTemplateVersion]`* — Verwendet eine vorhandene Startvorlage. Optional können Sie eine Vorlagenversion angeben. Wenn Sie keine Vorlagenversion angeben, verwendet das Skript die Standardversion.	Nein	Zeichenfolge	`RECOMMENDED`

Parameter	Beschreibung	Erforderlich	Typ	Standardwert
`--system-updates`	Definiert, ob Kernel- und Paket-Updates durchgeführt werden sollen, wenn die Instance gestartet wird. Gültige Werte sind: • `ALL_UPDATES` — Führt Systemaktualisierungen für den Kernel und die Pakete durch, wenn die Instance gestartet wird. • `NO_UPDATES` - Führt keine Systemaktualisierungen durch, wenn die Instance gestartet wird. • `MATCH_LAYER_SETTINGS` — Verwendet die `InstallUpdatesOnBoot` Eigenschaften der OpsWorks Ebene oder Instanz, um zu bestimmen, ob Systemupdates installiert werden sollen.	Nein	Zeichenfolge	`ALL_UPDATES`
`--http-username`	Der Name des Systems Manager `SecureString` Manager-Parameters, der den Benutzernamen speichert, der für die Authentifizierung im HTTP-Archiv verwendet wird, das die benutzerdefinierten Kochbücher enthält.	Nein	Zeichenfolge	

Parameter	Beschreibung	Erforderlich	Typ	Standardwert
`--http-password`	Der Name des Systems Manager `SecureString` Manager-Parameters, der das Passwort speichert, das für die Authentifizierung im HTTP-Archiv verwendet wird, das die benutzerdefinierten Kochbücher enthält.	Nein	Zeichenfolge	
`--repo-private-key`	Der Name des Systems Manager `SecureString` Manager-Parameters, der den SSH-Schlüssel speichert, der zur Authentifizierung bei dem Repository verwendet wird, das die benutzerdefinierten Kochbücher enthält. Wenn das Repository aktiviert ist GitHub, müssen Sie einen neuen `Ed25519` SSH-Schlüssel generieren. Wenn Sie keinen neuen `Ed25519` SSH-Schlüssel generieren, schlägt die Verbindung zum GitHub Repository fehl.	Nein	Zeichenfolge	

Parameter	Beschreibung	Erforderlich	Typ	Standardwert
`--lb-type`	Der Typ des Load Balancers, falls vorhanden, der bei der Migration Ihres vorhandenen Load Balancers erstellt werden soll. Gültige Werte sind: • `ALB`(Application Load Balancer) • `Classis`(Classic Load Balancer) • `None`(wenn Sie keinen Load Balancer erstellen möchten)	Nein	Zeichenfolge	`ALB`
`--lb-access-logs-path`	Der Pfad zu einem vorhandenen S3-Bucket und das Präfix zum Speichern der Load Balancer-Zugriffsprotokolle. Der S3-Bucket und der Load Balancer müssen sich in derselben Region befinden. Wenn Sie keinen Wert angeben und der `--lb-type` Parameterwert auf gesetzt ist`None`, erstellt das Skript einen neuen S3-Bucket und ein neues Präfix. Stellen Sie sicher, dass es eine geeignete Bucket-Richtlinie für dieses Präfix gibt.	Nein	Zeichenfolge	

Parameter	Beschreibung	Erforderlich	Typ	Standardwert
`--enable-instance-protection`	Wenn auf gesetzt`TRUE`, erstellt das Skript eine benutzerdefinierte Kündigungsrichtlinie (Lambda-Funktion) für Ihre Auto Scaling Scaling-Gruppe. EC2-Instances mit einem `protected_instance` Tag sind vor Scale-In-Ereignissen geschützt. Fügen Sie jeder EC2-Instance, die Sie vor Scale-In-Ereignissen schützen möchten, ein `protected_instance` Tag hinzu.	Nein	Boolesch	`FALSE`
`--command-logs-bucket`	Der Name eines vorhandenen S3-Buckets, in dem die `AWS ApplyChefRecipe` und - Protokolle gespeichert werden sollen. `MountEBSVolumes` Wenn Sie keinen Wert angeben, erstellt das Skript einen neuen S3-Bucket.	Nein	Zeichenfolge	`aws-opsworks-application-manager-logs-`*`account-id`
`--custom-json-bucket`	Der Name eines vorhandenen S3-Buckets zum Speichern von benutzerdefiniertem JSON. Wenn Sie keinen Wert angeben, erstellt das Skript einen neuen S3-Bucket.	Nein	Zeichenfolge	`aws-apply-chef-application-manager-transition-data-`*`account-id`

Hinweise:

- Wenn Sie ein privates GitHub Repository verwenden, müssen Sie einen neuen `Ed25519` Hostschlüssel für SSH erstellen. Dies liegt daran, dass GitHub geändert wurde, welche Schlüssel in SSH unterstützt werden, und das unverschlüsselte Git-Protokoll entfernt wurde. Weitere Informationen zum `Ed25519` Hostschlüssel findest du im GitHub Blogbeitrag Improving Git protocol security on GitHub. Nachdem Sie einen neuen `Ed25519` Hostschlüssel generiert haben, erstellen Sie einen Systems Manager `SecureString` Manager-Parameter für den SSH-Schlüssel und verwenden Sie den `SecureString` Parameternamen als Wert für den `--repo-private-key` Parameter. Weitere Informationen zum Erstellen eines Systems Manager `SecureString` Manager-Parameters finden Sie unter Erstellen eines SecureString Parameters (AWS CLI) oder Erstellen eines Systems Manager Manager-Parameters (Konsole) im AWS Systems Manager Benutzerhandbuch.
- Die `--repo-private-key` Parameter`--http-username`, `--http-password` und beziehen sich auf den Namen eines Systems Manager `SecureString` Manager-Parameters. Das Migrationsskript verwendet diese Parameter, wenn Sie das `AWS-ApplyChefRecipes` Dokument ausführen.
- Der `--http-username` Parameter erfordert, dass Sie auch einen Wert für den `--http-password` Parameter angeben.
- Für den `--http-password` Parameter müssen Sie auch einen Wert für den `--http-username` Parameter angeben.
- Legen Sie keine Werte für `--http-password` sowohl als auch fest`--repo-private-key`. Geben Sie entweder den Systems Manager `SecureString` Manager-Parameternamen eines SSH-Schlüssels (`--repo-private-key`) oder einen Repository-Benutzernamen (`--http-username`) und ein Passwort (`--http-password`) an.

Schritt 4: Führen Sie das Skript aus

Bei der Ausführung `python3 stack_exporter.py` können Sie entweder die Anwendung bereitstellen oder eine Startvorlage erstellen, indem Sie den Wert des `--provision-application` Parameters auf setzen`FALSE`.

Beispiel 1: Bereitstellen einer Systems Manager Application Manager-Anwendung

Mit dem folgenden Befehl werden Informationen zu einer vorhandenen OpsWorks Ebene abgerufen und eine Anwendung mithilfe der neueren OpsWorks Architektur bereitgestellt. Dadurch wird ein ähnliches Ergebnis erzielt wie die für den Stack konfigurierte Chef-Version. Das Skript stellt alle

erforderlichen Ressourcen, z. B. Auto Scaling Scaling-Gruppen CloudFormation, mithilfe bereit und registriert die Anwendung anschließend im Systems Manager Application Manager.

Ersetzen Sie *Stack-Region* und *Layer-ID* durch die Werte für Ihren Stack und Layer. OpsWorks

```
python3 stack_exporter.py \
     --layer-id layer-id \
     --region stack-region
```

Beispiel 2: Generieren Sie eine Vorlage

Mit dem folgenden Befehl werden Informationen zu einer vorhandenen OpsWorks Ebene abgerufen und eine CloudFormation Vorlage generiert. Wenn die Vorlage bereitgestellt wird, erzielt sie ein ähnliches Ergebnis wie die Verwendung von Chef 14. In diesem Beispiel werden keine Ressourcen bereitgestellt, da der `--provision-application` Parameter auf gesetzt ist. `FALSE`

Ersetzen Sie *stack-region* und *layer-id* durch die Werte für Ihren Stack und Ihre Ebene. OpsWorks

```
python3 stack_exporter.py \
    --layer-id layer-id \
    --region stack-region \
    --provision-application FALSE
```

Nachdem Sie den Befehl ausgeführt haben, können Sie die Vorlage in der Application Manager-Vorlagenbibliothek in Systems Manager überprüfen und die Vorlage auch bereitstellen. Weitere Informationen zum Anzeigen der Vorlagenbibliothek finden Sie im AWS Systems Manager Benutzerhandbuch unter Arbeiten mit der Vorlagenbibliothek.

Schritt 5: Einen CloudFormation Stack bereitstellen

> **Note**
>
> Sie müssen diesen Schritt nur abschließen, wenn Sie den `--provision-application` Parameter für das Skript auf setzen`FALSE`.

Wenn Sie den `--provision-application` Parameter mit dem Wert von angeben`FALSE`, enthält die Skriptausgabe den Namen und die URL für die CloudFormation Vorlage. Diese Vorlage stellt

einen vorgeschlagenen Ersatz für Ihren vorhandenen OpsWorks Stack und Ihre vorhandene Ebene dar.

Sie können die Vorlage mithilfe der Application Manager-Vorlagenbibliothek (empfohlen) oder mithilfe von bereitstellen CloudFormation. Weitere Informationen zur Arbeit mit der Vorlagenbibliothek finden Sie unter Arbeiten mit der Vorlagenbibliothek im AWS Systems Manager Benutzerhandbuch.

Schritt 6: Überprüfen Sie die bereitgestellten Ressourcen

Sie sind jetzt bereit, die bereitgestellten Ressourcen zu überprüfen.

1. Überprüfen Sie die Ressourcen für den bereitgestellten Stack mithilfe der AWS CloudFormation Konsole.

   a. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation und wählen Sie Stacks.

   b. Wählen Sie auf der Seite Stacks den Stack und dann den Tab Resources aus.

   c. Überprüfe auf der Registerkarte Ressourcen die aufgelisteten Ressourcen für deinen Stack. Die Liste der Ressourcen enthält eine EC2 Auto Scaling Scaling-Gruppe, die Sie in der Auto Scaling-Konsole überprüfen können, oder AWS CLI.

2. Überprüfen Sie die Ressourcen für die Anwendung mithilfe von Systems Manager Application Manager.

   a. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

   b. Wählen Sie im Navigationsbereich Application Manager aus.

   c. Wählen Sie im Abschnitt Anwendungen die benutzerdefinierte Anwendung aus. Der Anwendungsmanager öffnet die Registerkarte Übersicht.

   d. Wählen Sie die Registerkarte Resources (Ressourcen) aus. Auf der Registerkarte Ressourcen werden alle Ressourcen angezeigt, die für Ihren OpsWorks Stack und Ihre Ebene migriert wurden. Der Anwendungsname beinhaltet den Namen des OpsWorks Stacks und ist als *App-Stack-Name-Suffix* formatiert, wobei das *Suffix* für die ersten sechs *Zeichen* der Stack-ID steht. Weitere Informationen zum Anzeigen von Ressourcen in Application Manager finden Sie unter Anzeigen von Anwendungsressourcen im Benutzerhandbuch.AWS Systems Manager

Schritt 7: Starten Sie eine Instanz

Nachdem Sie eine Instanz bereitgestellt haben, können Sie die Instanz testen. Zu diesem Zeitpunkt werden keine Instanzen ausgeführt.

Um Ihre Instances online zu schalten, passen Sie die `Desired capacity` Werte für `Min`, `Max` und für die Auto Scaling Scaling-Gruppe auf eine Zahl an, die für Ihre Anwendung sinnvoll ist. Anfänglich können Sie diese Werte auf 1 setzen, um eine einzelne Instance online zu schalten und zu überprüfen, ob die Instance alle erwarteten Aktionen ausführt, einschließlich der Ausführung Ihrer benutzerdefinierten Chef-Rezepte.

Schritt 8: Überprüfen Sie die Instanz

Nachdem Sie eine Instance gestartet haben, stellen Sie sicher, dass sie wie erwartet ausgeführt wird.

1. Überprüfen Sie den Chef `startup` und die `terminate` Protokolle, die sich im S3-Bucket befinden, der durch den `--command-logs-bucket` Parameter des Skripts angegeben ist. Standardmäßig werden die Protokolle in einem Bucket mit dem Namen gespeichert`aws-opsworks-application-manager-logs-`*`account-id`*.

   a. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/s3/.

   b. Wählen Sie den Bucket aus, der Ihre Logs enthält.

   c. Navigieren Sie zum `ApplyChefRecipes` Präfix, um Ihre Logs einzusehen.

2. Überprüfen Sie die Konnektivität und den Zustand des Application Load Balancers.

   Gehen Sie wie folgt vor, um die Zugriffsprotokolle für Ihren Load Balancer einzusehen. Mithilfe des Skriptparameters können Sie den S3-Bucket angeben, in dem Sie die Load Balancer-Zugriffsprotokolle speichern möchten. `--lb-access-logs-path`

   a. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/s3/.

   b. Wählen Sie Ihren S3-Bucket aus und navigieren Sie dann zu dem Präfix, das Ihre Logs enthält.

3. Stellen Sie sicher, dass die Instance alle Zustandsprüfungen von Auto Scaling und Application Load Balancer besteht (falls Sie welche konfiguriert haben).

   Informationen zum Zustand von Auto Scaling finden Sie auf der neuen Registerkarte Instances.

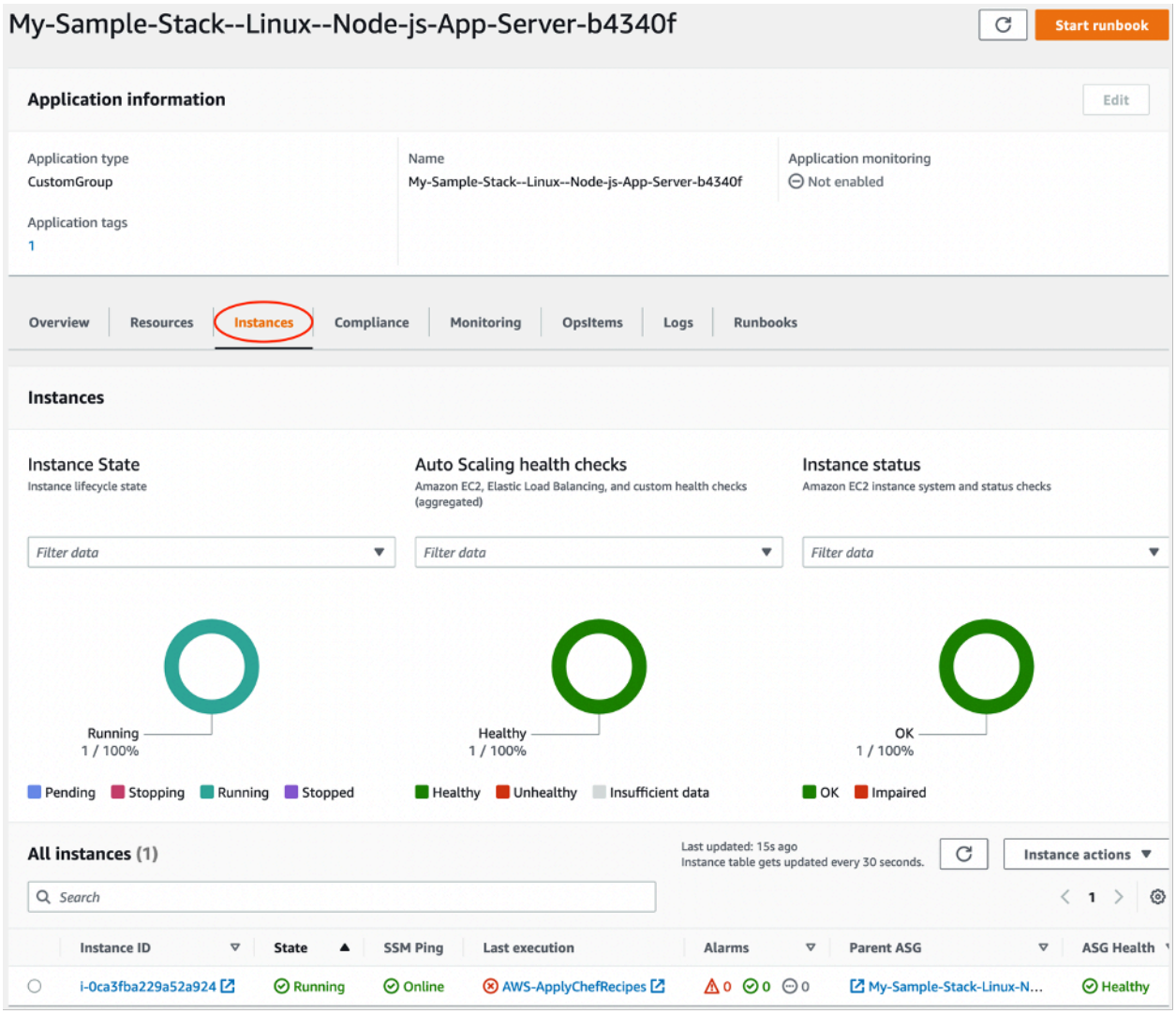
a. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
b. Wählen Sie im Navigationsbereich Application Manager aus.
c. Wählen Sie im Abschnitt Anwendungen die Option Benutzerdefinierte Anwendungen aus.
d. Wählen Sie die Anwendung in der Liste aus. Der Anwendungsmanager öffnet die Registerkarte „Übersicht“.
e. Wählen Sie die Registerkarte Instances, um Informationen zum Zustand von Auto Scaling anzuzeigen.

Nachdem Sie sich vergewissert haben, dass die Chef-Rezepte erfolgreich ausgeführt wurden, können Sie die Kapazität der Auto Scaling Scaling-Gruppe verringern, um die Instance zu beenden. Wenn Sie über benutzerdefinierte Kündigungsrezepte verfügen, stellen Sie sicher, dass die Rezepte erwartungsgemäß funktionieren.

Schritt 9: Überwachen und Ausführen von Vorgängen auf Ihren Instances mithilfe des Systems Manager Application Manager

Sie können jetzt Operationen auf Ihren Instances mithilfe einer neuen Registerkarte „Instances“ auf der Application Manager-Seite überwachen und ausführen. Weitere Informationen zur Arbeit mit der Registerkarte „Instances“ finden Sie im AWS Systems Manager Benutzerhandbuch unter Arbeiten mit Ihren Anwendungsinstanzen.

Sie können die Registerkarte „Instanzen“ verwenden, um mehrere AWS Instanzen an einem Ort anzuzeigen. Auf dieser Registerkarte können Sie Informationen zum Zustand der Instance einsehen und Probleme beheben.



Gehen Sie wie folgt vor, um den Tab Instances aufzurufen.

1. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie im Abschnitt Anwendungen die Option Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Der Anwendungsmanager öffnet die Registerkarte „Übersicht“.
5. Wählen Sie die Registerkarte Instances, um Informationen zum Status Ihrer Instance und zum EC2-Status anzuzeigen.

Häufig gestellte Fragen

Die folgenden FAQs geben Antworten auf einige häufig gestellte Fragen.

Themen

Welche AWS OpsWorks Stacks Versionen kann ich migrieren?

Sie können nur Chef 11.10- und Chef 12-, Amazon Linux-, Amazon Linux 2-, Ubuntu- und Red Hat Enterprise Linux 7-Stacks migrieren.

Welche Chef-Versionen können meine migrierten Instances verwenden?

Migrierte Instanzen können die Chef-Versionen 11 bis 14 verwenden.

> **Note**
>
> Die Windows-Stack-Migration wird nicht unterstützt.

Welche Repository-Typen kann ich migrieren?

Sie können die Repository-Typen S3, Git und HTTP migrieren.

Kann ich weiterhin ein privates Git-Repository verwenden?

Ja, du kannst weiterhin ein privates Git-Repository verwenden.

Wenn du ein privates GitHub Repository verwendest, musst du einen neuen `Ed25519` Hostschlüssel für SSH erstellen. Dies liegt daran, dass GitHub geändert wurde, welche Schlüssel in SSH unterstützt werden, und das unverschlüsselte Git-Protokoll entfernt wurde. Weitere Informationen zum `Ed25519` Hostschlüssel findest du im GitHub Blogbeitrag Improving Git protocol security on GitHub. Nachdem Sie einen neuen `Ed25519` Hostschlüssel generiert haben, erstellen Sie einen Systems Manager `SecureString` Manager-Parameter für diesen SSH-Schlüssel und verwenden Sie den Parameternamen als Wert für den `--repo-private-key` Parameter. Weitere Informationen zum Erstellen eines Systems Manager `SecureString` Manager-Parameters finden Sie unter Create a SecureString parameter (AWS CLI) im AWS Systems Manager Benutzerhandbuch.

Erstellen Sie für jeden anderen Git-Repository-Typ einen Systems Manager `SecureString` Manager-Parameter für diesen SSH-Schlüssel und verwenden Sie den Parameternamen als Wert für den `--repo-private-key` Parameter des Skripts.

Mit welchen SSH-Schlüsseln kann ich auf meine Instanzen zugreifen?

Wenn Sie das Skript ausführen, migriert das Skript die im Stack konfigurierten SSH-Schlüssel und -Instanzen. Sie können die SSH-Schlüssel verwenden, um auf Ihre Instanz zuzugreifen. Wenn SSH-Schlüssel für den Stack und die Instanz bereitgestellt werden, verwendet das Skript die Schlüssel aus dem Stack. Wenn Sie sich nicht sicher sind, welche SSH-Schlüssel Sie verwenden sollen, sehen Sie sich die Instances in der EC2-Konsole an (https://console.aws.amazon.com/ec2/). Auf der Detailseite in der EC2-Konsole werden die SSH-Schlüssel für Ihre Instance angezeigt.

Warum werden meine Instances automatisch ein- und ausskaliert?

Auto Scaling skaliert Instances auf der Grundlage der Skalierungsregeln für die Auto Scaling Scaling-Gruppe. Sie können die Kapazitätswerte Min., Max und Desired für Ihre Gruppe festlegen. Die Auto Scaling Scaling-Gruppe skaliert Ihre Kapazität automatisch entsprechend, wenn Sie diese Werte aktualisieren.

Kann ich Auto Scaling ausschalten?

Sie können Auto Scaling deaktivieren, indem Sie die Werte Min., Max und Gewünschte Kapazität der Auto Scaling Scaling-Gruppe auf dieselbe Zahl setzen. Wenn Sie beispielsweise immer über zehn Instances verfügen möchten, legen Sie die Werte für Min., Max und Gewünschte Kapazität auf zehn fest.

Kann ich Kernel- und Paket-Updates auf gestarteten EC2-Instances durchführen?

Standardmäßig erfolgen Kernel- und Paket-Updates, wenn die EC2-Instance gestartet wird. Gehen Sie wie folgt vor, um Kernel- oder Paket-Updates auf einer gestarteten EC2-Instance durchzuführen. Beispielsweise möchten Sie möglicherweise Updates anwenden, nachdem Sie die Rezepte „deploy“ oder „configure“ ausgeführt haben.

1. Stellen Sie eine Verbindung zu Ihrer EC2- Instance her.
2. Erstellen Sie die folgende `perform_upgrade` Funktion und führen Sie sie auf Ihrer Instanz aus.

```
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. Nach den Kernel- und Paket-Updates müssen Sie möglicherweise Ihre EC2-Instance neu starten. Um zu überprüfen, ob ein Neustart erforderlich ist, erstellen Sie die folgende `reboot_if_required` Funktion und führen Sie sie auf Ihrer EC2-Instance aus.

```
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. Wenn die `reboot_if_required` Ergebnisse in einer `reboot is required` Meldung ausgeführt werden, starten Sie die EC2-Instance neu. Wenn Sie eine `reboot is not required` Nachricht erhalten, müssen Sie die EC2-Instance nicht neu starten.

Warum enthalten die EBS-Volumes in meinen Instances keine Daten?

Wenn Sie das Skript ausführen, migriert das Skript die Konfiguration der EBS-Volumes und erstellt so eine Ersatzarchitektur für Ihren OpsWorks Stack und Ihre Ebene. Das Skript migriert weder die tatsächlichen Instanzen noch die in den Instanzen enthaltenen Daten. Das Skript migriert nur die Konfiguration von EBS-Volumes auf Layer-Ebene und hängt die leeren EBS-Volumes an gestartete EC2-Instances an.

Gehen Sie wie folgt vor, um Daten aus den EBS-Volumes Ihrer vorherigen Instances abzurufen.

1. Erstellen Sie einen Snapshot der EBS-Volumes Ihrer vorherigen Instances. Weitere Informationen zum Erstellen eines Snapshots finden Sie unter Amazon EBS-Snapshot erstellen im Amazon EC2 EC2-Benutzerhandbuch.
2. Erstellen Sie ein Volume aus Ihrem Snapshot. Weitere Informationen zum Erstellen eines Volumes aus einem Snapshot finden Sie unter Create a Volume from a Snapshot im Amazon EC2 EC2-Benutzerhandbuch.

3. Hängen Sie das von Ihnen erstellte Volume an die Instances an. Weitere Informationen zum Anhängen von Volumes finden Sie unter Anhängen eines Amazon EBS-Volumes an eine Instance im Amazon EC2 EC2-Benutzerhandbuch.

Warum wurden die in meiner Startvorlage beschriebenen EBS-Volumes nicht bereitgestellt?

Wenn Sie eine Startvorlagen-ID für den `--launch-template` Parameter mit EBS-Volumes angeben, hängt das Skript die EBS-Volumes an, mountet die Volumes jedoch nicht. Sie können die angehängten EBS-Volumes mounten, indem Sie das `MountEBSVolumes` RunCommand Dokument ausführen, das das Skript für die gestartete EC2-Instance erstellt hat.

Wenn Sie keinen `--launch-template` Parameter festlegen, erstellt das Skript eine Vorlage. Wenn die Auto Scaling Scaling-Gruppe eine neue EC2-Instance startet, hängt die Auto Scaling Scaling-Gruppe automatisch die EBS-Volumes an und führt dann den `SetupAutomation` Befehl aus, um die angehängten Volumes an den in den Layer-Einstellungen konfigurierten Mountpunkten zu mounten.

Wo finde ich die Volume Logs von Chef Recipe und Mount EBS?

OpsWorks übermittelt die Protokolle an einen S3-Bucket, den Sie angeben können, indem Sie einen Wert für den `--command-logs-bucket` Parameter angeben. Der Standardname des S3-Buckets hat das Format:`aws-opsworks-stacks-application-manager-logs-`*`account-id`*. Chef-Rezeptprotokolle werden im `ApplyChefRecipes` Präfix gespeichert. Mount EBS-Volumenprotokolle werden im `MountEBSVolumes` Präfix gespeichert. Alle Ebenen, die aus einem Stack migriert werden, liefern Protokolle an denselben S3-Bucket.

> **Note**
>
> - Die Lifecycle-Konfiguration des S3-Buckets umfasst eine Regel zum Löschen der Protokolle nach 30 Tagen. Wenn Sie die Protokolle länger als 30 Tage aufbewahren möchten, müssen Sie die Regel in der Lifecycle-Konfiguration des S3-Buckets aktualisieren.
> - Derzeit werden OpsWorks nur Chef `setup` und `terminate` Rezepte protokolliert.

Wo finde ich das Debug-Protokoll für das Migrationsskript?

Das Skript platziert Debug-Logs in einem Bucket mit dem Namen. `aws-opsworks-stacks-transition-logs-`*`account-id`* Sie finden die Debug-Protokolle im `migration_script` Ordner des S3-Buckets unter Ordnern, die dem Namen der Ebene entsprechen, die Sie migriert haben.

Unterstützt das Migrationsskript die Versionierung von CloudFormation Vorlagen?

Das Skript generiert Systems Manager Manager-Dokumente des Typs CloudFormation , die einen Ersatz für die Ebene oder den Stapel bilden, den Sie migrieren möchten. Wenn Sie das Skript erneut ausführen, selbst mit denselben Parametern, wird eine neue Version der zuvor exportierten Layer-Vorlage exportiert. Die Vorlagenversionen werden im selben S3-Bucket wie die Skriptprotokolle gespeichert.

Kann ich mehrere Ebenen migrieren?

Der `--layer-id` Parameter des Skripts wird in einer einzigen Ebene übergeben. Um mehrere Ebenen zu migrieren, führen Sie das Skript erneut aus und übergeben Sie eine andere`--layer-id`.

Ebenen, die Teil desselben OpsWorks Stacks sind, werden in Application Manager unter derselben Anwendung aufgeführt.

1. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie im Abschnitt Anwendungen die Option Benutzerdefinierte Anwendungen aus.
4. Wählen Sie Ihre Anwendung. Der Name der Anwendung beginnt mit`app-`*`stack-name`*`-`*`first-six-characters-stack-id`*.
5. Das Element der obersten Ebene, das mit App beginnt, zeigt alle Komponenten, die Ihrem OpsWorks Stack entsprechen. Dazu gehören Komponenten, die Ihrer OpsWorks Ebene entsprechen.
6. Wählen Sie die Komponente aus, die der Ebene entspricht, um die Ressourcen für die Ebene anzuzeigen. Die Komponenten, die OpsWorks Ebenen darstellen, sind auch im Bereich Benutzerdefinierte Anwendungen als einzelne Anwendungen sichtbar.

Wie erstelle ich einen `SecureString` Parameter?

Sie können Systems Manager verwenden, um einen `SecureString` Parameter zu erstellen. Weitere Informationen zum Erstellen eines Systems Manager `SecureString` Manager-Parameters finden Sie unter Erstellen eines SecureString Parameters (AWS CLI) oder Erstellen eines Systems Manager Manager-Parameters (Konsole) im AWS Systems Manager Benutzerhandbuch.

Sie müssen einen `SecureString` Parameter als Wert für die `--repo-private-key` Parameter `--http-username--http-password`, oder angeben.

Wie kann ich Instances in der neuen Auto Scaling Scaling-Gruppe vor Terminierungsereignissen schützen?

Sie können Instances schützen, indem Sie den `--enable-instance-protection` Parameter auf festlegen `TRUE` und jeder EC2-Instance, die Sie vor Kündigungsereignissen schützen möchten, einen `protected_instance` Tag-Schlüssel hinzufügen. Wenn Sie den `--enable-instance-protection` Parameter auf setzen `TRUE` und einen `protected_instance` Tag-Schlüssel hinzufügen, fügt das Skript Ihrer neuen Auto Scaling Scaling-Gruppe eine benutzerdefinierte Kündigungsrichtlinie hinzu und unterbricht den `ReplaceUnhealthy` Prozess. Instances mit dem `protected_instance` Tag-Schlüssel sind vor den folgenden Terminierungsereignissen geschützt:

- Skalierung von Ereignissen
- Instance-Aktualisierung
- Neuausgleich
- Maximale Lebensdauer der Instanz
- Kündigung der Listing-Instance zulassen
- Kündigung und Ersatz fehlerhafter Instances

> **Note**
>
> Sie müssen den `protected_instance` Tag-Schlüssel für Instances festlegen, die Sie schützen möchten. Beim Tag-Schlüssel wird zwischen Groß- und Kleinschreibung unterschieden. Jede Instanz mit diesem Tag-Schlüssel ist unabhängig vom Tag-Wert geschützt.
> Um die Laufzeit der benutzerdefinierten Kündigungsrichtlinie zu reduzieren, können Sie die Standardanzahl von Instanzen erhöhen, die die Lambda-Funktion verwendet, um nach geschützten Instanzen zu filtern, indem Sie den Wert für die `default_sample_size`

Funktionscodevariable aktualisieren. Der Standardwert ist 15. Wenn Sie den erhöhen`default_sample_size`, müssen Sie möglicherweise den der Lambda-Funktion zugewiesenen Speicher erhöhen, was die Kosten Ihrer Lambda-Funktion erhöhen würde. Informationen zu AWS Lambda -Preisen erhalten Sie unter AWS Lambda Pricing (Preise für WAF).

Welche Load Balancer sind mit dem Migrationsskript verfügbar?

Das Skript bietet drei Load Balancer-Optionen.

- (Empfohlen) Erstellen Sie einen neuen Application Load Balancer. Standardmäßig erstellt das Skript einen neuen Application Load Balancer. Sie können den `--lb-type` Parameter auch auf setzen. `ALB` Weitere Informationen zu Application Load Balancers finden Sie unter Was ist ein Application Load Balancer? im Elastic Load Balancing User Guide.
- Wenn ein Application Load Balancer keine Option ist, erstellen Sie einen Classic Load Balancer, indem Sie den `--lb-type` Parameter auf setzen. `Classic` Wenn Sie diese Option auswählen, wird Ihr vorhandener Classic Load Balancer, der mit Ihrem OpsWorks Layer verbunden ist, von Ihrer Anwendung getrennt. Weitere Informationen zu Application Load Balancers finden Sie unter Was ist ein Classic Load Balancer? im Elastic Load Balancing: Classic Load Balancers Benutzerhandbuch.
- Sie können einen vorhandenen Load Balancer anhängen, indem Sie den `--lb-type` Parameter auf setzen. `None`

> **Important**
>
> Wir empfehlen, neue Elastic Load Balancing Load Balancer für Ihre AWS OpsWorks Stacks-Ebenen zu erstellen. Wenn Sie sich dafür entscheiden, einen vorhandenen Elastic Load Balancing Load Balancer zu verwenden, sollten Sie zunächst sicherstellen, dass er nicht für andere Zwecke verwendet wird und keine angehängten Instances hat. Nachdem der Load Balancer an die Ebene angehängt wurde, werden alle vorhandenen Instances OpsWorks entfernt und der Load Balancer so konfiguriert, dass er nur die Instances der Ebene verarbeitet. Es ist zwar technisch möglich, die Elastic Load Balancing Balancing-Konsole oder API zu verwenden, um die Konfiguration eines Load Balancers zu ändern, nachdem er an eine Ebene angehängt wurde, aber Sie sollten dies nicht tun, da die Änderungen nicht dauerhaft sind.

So fügen Sie Ihren vorhandenen OpsWorks Layer-Load Balancer Ihrer Auto Scaling Scaling-Gruppe hinzu

1. Führen Sie das Migrationsskript aus, wobei der `--lb-type` Parameter auf `None` gesetzt ist. Wenn der Wert auf gesetzt ist`None`, klont oder erstellt das Skript keinen Load Balancer.
2. Nachdem das Skript den CloudFormation Stack bereitgestellt hat, aktualisieren Sie die Auto Scaling Scaling-Gruppen `Min` `Max` und `Desired capacity` -Werte und testen Sie dann Ihre Anwendung.
3. Wählen Sie die in `Link to the template` der Skriptausgabe angezeigte Option aus. Wenn Sie Ihr Terminal geschlossen haben, gehen Sie wie folgt vor, um auf die Vorlage zuzugreifen.
   a. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
   b. Wählen Sie im Navigationsbereich Application Manager aus.
   c. Wählen Sie CloudFormation Stacks und dann Template-Bibliothek aus.
   d. Wählen Sie Owned by me und suchen Sie nach Ihrer Vorlage.
4. Wählen Sie in der CloudFormation Vorlage im Menü Aktionen die Option Bearbeiten aus.
5. Aktualisieren Sie die `LabelBalancerNames` Eigenschaft im `ApplicationAsg` Ressourcenbereich der CloudFormation Vorlage.

```
ApplicationAsg:
   DependsOn: CustomTerminationLambdaPermission
   Properties:
   #(other properties in ApplicationAsg to remain unchanged)
      LoadBalancerNames:
        - load-balancer-name
      HealthCheckType: ELB
```

6. Wenn Sie möchten, dass die Zustandsprüfung Ihrer Auto Scaling Scaling-Gruppeninstanzen auch die Integritätsprüfung des Load Balancers verwendet, entfernen Sie den folgenden Abschnitt `HealthCheckType` und geben Sie ein`ELB`. Wenn Sie nur EC2-Integritätsprüfungen benötigen, müssen Sie die Vorlage nicht ändern.
7. Speichern Sie Ihre Änderungen. Beim Speichern wird eine neue Standardversion der Vorlage erstellt. Wenn Sie das Skript für den Layer zum ersten Mal ausführen und Änderungen zum ersten Mal in der Konsole gespeichert haben, ist die neuere Version 2.
8. Wählen Sie unter Aktionen die Option Bereitstellungsstapel aus.

9. Bestätigen Sie, dass Sie die Standardversion der Vorlage verwenden möchten. Vergewissern Sie sich, dass Select a existing stack ausgewählt ist und wählen Sie den CloudFormation Stack aus, der aktualisiert werden soll.
10. Wählen Sie für jede der nachfolgenden Seiten Weiter aus, bis Sie die Seite „Überprüfen und Bereitstellen“ sehen. Wählen Sie auf der Seite Überprüfen und Bereitstellen sowohl Ich bestätige, dass AWS CloudFormation möglicherweise IAM-Ressourcen mit benutzerdefinierten Namen erstellt werden, als auch die Option Ich verstehe, dass Änderungen an der ausgewählten Vorlage AWS CloudFormation dazu führen können, dass vorhandene AWS Ressourcen aktualisiert oder entfernt werden.
11. Wählen Sie Stack bereitstellen.

Wenn Sie Ihre Aktualisierungen rückgängig machen müssen, gehen Sie wie folgt vor.

1. Wählen Sie Aktionen und dann Bereitstellungsstapel aus.
2. Wählen Sie Wählen Sie eine der vorhandenen Versionen aus und wählen Sie dann die vorherige Vorlagenversion aus.
3. Wählen Sie „Einen vorhandenen Stapel auswählen“ und dann den zu CloudFormation aktualisierenden Stapel aus.

Werden Rezepte zur Konfiguration benutzerdefinierter Kochbücher migriert?

Die Ausführung benutzerdefinierter Kochbücher während eines Setup-Ereignisses wird nicht unterstützt. Das Skript migriert benutzerdefinierte Kochbuch-Konfigurationsrezepte und erstellt ein Systems Manager Automation-Runbook für Sie. Sie müssen die Rezepte jedoch manuell ausführen.

Führen Sie die folgenden Schritte aus, um Ihre Konfigurationsrezepte auszuführen.

1. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie im Abschnitt Anwendungen die Option Benutzerdefinierte Anwendungen aus.
4. Wählen Sie Ihre Anwendung. Der Name der Anwendung beginnt mit`app-`*`stack-name`*.
5. Wählen Sie Resources und dann das Configure Runbook aus.
6. Wählen Sie „Automatisierung ausführen“.

7. Wählen Sie die Instanz-IDs aus, für die Sie die Konfigurationsrezepte ausführen möchten, und wählen Sie dann Execute.

Kann ich Rezepte zum Bereitstellen und Aufheben der Bereitstellung auf meinen neu erstellten Instances ausführen?

Das Skript kann je nach Konfiguration Ihres Layers drei mögliche Automation-Runbooks erstellen.

- Aufstellen
- Konfiguration
- Beenden

Das Skript kann auch die folgenden Systems Manager Manager-Parameter erstellen, die Eingabewerte für das `AWS-ApplyChefRecipes Run Command` Dokument enthalten.

- Aufstellen
- Bereitstellen
- Konfiguration
- Bereitstellung aufheben
- Beenden

Wenn ein Scale-Out-Ereignis eintritt, wird das Runbook für die Setup-Automatisierung automatisch ausgeführt. Dazu gehören die Einrichtung und Bereitstellung von benutzerdefinierten Kochbuchrezepten aus Ihrer ursprünglichen Ebene. OpsWorks Wenn ein Scale-In-Ereignis eintritt, wird das Terminate Automation-Runbook automatisch ausgeführt. Das Terminate Automation-Runbook enthält die Shutdown-Rezepte aus Ihrem ursprünglichen Layer. OpsWorks

Wenn Sie Rezepte manuell ausführen, bereitstellen oder konfigurieren möchten, gehen Sie wie folgt vor.

1. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie im Abschnitt Anwendungen die Option Benutzerdefinierte Anwendungen aus.

4. Wählen Sie Ihre Anwendung. Der Name der Anwendung beginnt mit`app-`*`stack-name`*`-`*`first-six-characters-stack-id`*. Der Anwendungsmanager öffnet die Registerkarte Übersicht.
5. Wählen Sie Ressourcen und dann das Runbook zur Konfiguration von Automation aus.
6. Wählen Sie „Automatisierung ausführen“.
7. Verweisen Sie für den Eingabeparameter des `applyChefRecipesPropertiesParameter` Automatisierungs-Runbooks auf den richtigen Systems Manager Manager-Parameter. Der Name des Systems Manager Manager-Parameters folgt dem Format`/ApplyChefRecipes-Preset/`*`OpsWorks-stack-name`*`-`*`OpsWorks-layer-name`*`-`*`first-six-characters-stack-id`*`/`*`event`*, in dem der Wert für das *`Ereignis`* lautet `ConfigureDeploy`, oder `Undeploy` hängt von den Rezepten ab, die Sie ausführen möchten.
8. Wählen Sie die Instanz-IDs aus, auf denen Sie die Rezepte ausführen möchten, und klicken Sie auf Ausführen.

Kann ich ändern, über welche Subnetze sich meine Auto Scaling Scaling-Gruppe erstreckt?

Standardmäßig umfasst die Auto Scaling Scaling-Gruppe alle Subnetze in Ihrer OpsWorks Stack-VPC. Gehen Sie wie folgt vor, um zu aktualisieren, welche Subnetze sich erstrecken sollen.

1. Wählen Sie aus, wie es in der Ausgabe des Skripts `Link to the template` angezeigt wird. Wenn Sie Ihr Terminal geschlossen haben, gehen Sie wie folgt vor, um auf die Vorlage zuzugreifen.
   a. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
   b. Wählen Sie im Navigationsbereich Application Manager aus.
   c. Wählen Sie CloudFormation Stacks und dann Template-Bibliothek aus.
   d. Wählen Sie Owned by me und suchen Sie nach Ihrer Vorlage.
2. Wählen Sie unter Aktionen die Option Bereitstellungsstapel aus.
3. Bestätigen Sie, dass Sie die Standardvorlage verwenden möchten. Wählen Sie Bestehenden Stack auswählen und wählen Sie dann den CloudFormation Stack aus, der aktualisiert werden soll.

> **Note**
>
> Wenn Sie das Skript mit dem auf gesetzten `--provision-application` Parameter ausgeführt haben`FALSE`, müssen Sie einen neuen CloudFormation Stack erstellen.

4. Geben Sie für den `SubnetIDs` Parameter eine durch Kommas getrennte Liste der Subnetz-IDs an, über die sich Ihre Auto Scaling Scaling-Gruppe erstrecken soll.
5. Wählen Sie Weiter, bis die Seite „Überprüfen und Bereitstellen“ angezeigt wird.
6. Wählen Sie auf der Seite Überprüfen und Bereitstellen die Option Ich bestätige, dass AWS CloudFormation möglicherweise IAM-Ressourcen mit benutzerdefinierten Namen erstellt werden, und ich verstehe, dass Änderungen an der ausgewählten Vorlage AWS CloudFormation dazu führen können, dass vorhandene AWS Ressourcen aktualisiert oder entfernt werden.
7. Wählen Sie Stack bereitstellen.

Fehlerbehebung

Dieser Abschnitt enthält einige häufig auftretende Probleme und Lösungsvorschläge für diese Probleme.

Themen

Vorausgesetzt, das Prinzipal ist nicht gültig

Problem: Sie erhalten eine Fehlermeldung, die besagt, dass der von Ihnen angegebene Prinzipal nicht gültig ist.

Ursache: Dies liegt daran, dass die Auto Scaling Scaling-Gruppe keine Servicerolle hat.

Lösung: Erstellen Sie eine Auto Scaling Scaling-Gruppe in der Region, in der der Fehler aufgetreten ist. Durch das Erstellen einer Auto Scaling Scaling-Gruppe wird die erforderliche serviceverknüpfte Rolle für Ihre benutzerdefinierte Kündigungsrichtlinie erstellt.

Der CloudFormation Stack kann nicht gelöscht werden, wenn gruppengeschützte Auto Scaling Scaling-Instanzen aktiviert sind

Problem: Der `--enable-instance-protection` Parameter ist auf gesetzt `TRUE` und einige der EC2-Instances Ihrer Auto Scaling Scaling-Gruppe sind mit dem `protected_instance` Tag-Schlüssel geschützt, wodurch verhindert wird, dass Ihr AWS CloudFormation Stack vollständig gelöscht wird.

Ursache: Die EC2-Instances verfügen über einen `protected_instance` Tag-Schlüssel, der sie vor Terminierungsereignissen schützt.

Lösung: Entfernen Sie den `protected_instance` Tag-Schlüssel aus den EC2-Instances. Dadurch kann die Auto Scaling Scaling-Gruppe herunterskalieren. Nachdem die Auto Scaling Scaling-Gruppe herunterskaliert wurde, können Sie den AWS CloudFormation Stack löschen.

Fehler „Zugriff verweigert“ bei der Bereitstellung eines vorhandenen S3-Buckets und -Präfixes

Problem: Sie erhalten eine `AccessDenied` Fehlermeldung, wenn Sie einen vorhandenen S3-Bucket und ein Präfix angeben.

Ursache: Die S3-Bucket-Richtlinie bietet nicht die erforderlichen Berechtigungen, um die Load Balancer-Logs an den Bucket zu übermitteln.

Lösung: Aktualisieren Sie die S3-Bucket-Richtlinie, damit das Skript die Load Balancer-Zugriffsprotokolle an den Bucket übermitteln kann. Weitere Informationen zur Aktualisierung der Bucket-Richtlinie finden Sie unter Aktivieren von Zugriffsprotokollen für Ihren Application Load Balancer im Elastic Load Balancing: Application Load Balancers User Guide.

Verwenden des Werkzeugs „An Ort und Stelle AWS OpsWorks Stacks lösen“

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren.

In diesem Abschnitt wird beschrieben, wie Sie das Tool AWS OpsWorks Stacks Detach in Place verwenden, um Ihre OpsWorks Instances vom Stacks-Service zu trennen. OpsWorks

Die Instanzen, die Sie trennen, bleiben in Ihren AWS-Konto, aber Sie können sie nicht mehr mithilfe von. OpsWorks Stattdessen verwenden Sie Amazon EC2 oder einen anderen EC2-kompatiblen Ansatz AWS Systems Manager, um die Instances zu konfigurieren und zu verwalten.

Auf einer höheren Ebene umfasst der Prozess der Trennung die folgenden Schritte:

1. Das Tool führt Validierungsprüfungen durch, um sicherzustellen, dass die Ressourcen für die Trennung bereit sind.
2. Das Tool exportiert die benutzerdefinierte JSON-Datei aus Ihrem OpsWorks Stack und speichert sie als Objekt in Amazon S3.
3. Das Tool erstellt Systems Manager Automation-Dokumente, die jedes OpsWorks Stacks-Lifecycle-Ereignis darstellen.
4. Das Tool erstellt einen AWS Service Catalog AppRegistry Katalog für alle Instances, die getrennt werden, und trennt alle Elastic Load Balancing (ELB) -Load Balancer von den OpsWorks Layern.
5. Schließlich trennt und deregistriert das Tool andere Ressourcen, einschließlich Amazon Relational Database Service (Amazon RDS) -Instances.

Wie funktioniert der Prozess

Das Tool Detach In Place bietet die folgenden drei Befehle und ein assistentenähnliches Tool, das Sie durch eine Reihe von Schritten zur Überprüfung und Konfiguration Ihrer Instances führt, bevor Sie mit dem Trennen Ihres Layers fortfahren.

Befehl	Beschreibung
`handle-prerequisites`	Mit diesem Befehl wird analysiert, ob alle Instanzen in einer Ebene getrennt werden können, und es werden alle Voraussetzungen erfüllt. Die Instanzen müssen sich in einem fehlerfreien Zustand befinden OpsWorks, sie dürfen keine zeit- oder lastbasierten Auto Scaler haben und sie müssen die neueste OpsWorks Agent-Version installiert haben.

Befehl	Beschreibung
	Darüber hinaus überprüft der Befehl, ob alle Instanzen über die zur Unterstützung des SSM-Agenten erforderlichen Berechtigungen verfügen und ob die neueste SSM-Agent-Version installiert ist. Der Befehl installiert den SSM-Agenten, falls er nicht vorhanden ist, und aktualisiert den SSM-Agenten, wenn er nicht die neueste Version verwendet. Der Befehl fügt auch alle erforderlichen Berechtigungen hinzu.

Befehl	Beschreibung
`detach`	Mit diesem Befehl werden alle OpsWorks Instanzen für die angegebene Ebene getrennt. Zunächst führt der Befehl eine Prüfung der Voraussetzungen durch, um sicherzus tellen, dass die Ebene getrennt werden kann. Wenn Sie die Voraussetzungen nicht erfüllen möchten, haben Sie die Möglichkeit, das Trennen zu erzwingen. Als Nächstes gibt der Befehl an, dass alle Tags, die Ihren Instances durch OpsWorks Tagging-APIs oder durch die Weitergabe von Tags aus Ihren Layern und Stacks hinzugefü gt wurden, beibehalten werden. Sie können jedes dieser Tags mithilfe der entsprechenden EC2-APIs entfernen, nachdem die Trennung abgeschlossen ist. Anschließend prüft der Befehl, ob Sie die Chef-bezogene Konfiguration in SSM-Parameter exportieren möchten. Wenn Sie einen Classic Load Balancer an den Layer angeschlossen haben, fragt der Befehl, ob er den Load Balancer trennen kann, um Ausfallzeiten zu vermeiden.

Befehl	Beschreibung
`cleanup`	Dieser Befehl löscht alle Entitäten in OpsWorks Ihrem Konto. Er beendet die Instanzen und löscht alle Stacks. Dies sollte als letzter Schritt zur Bereinigung des Kontos für Ressource n verwendet werden, die nicht mehr benötigt werden. **Note** Wir empfehlen, dass Sie das neue Setup einige Tage lang ausführen , bevor Sie den `cleanup` Befehl ausführen. Dadurch wird sicherges tellt, dass alle erforderlichen Konfigura tionen aus dem Stack bei Bedarf sofort verfügbar sind.

Einschränkungen

Der Hauptzweck des Tools Detach In Place besteht darin, die OpsWorks Stacks-Instanzen sicher zu trennen. In diesem Abschnitt werden die Einschränkungen des Tools zusammengefasst.

- Windows SSM Agent — Wenn der SSM Agent nicht auf der Instanz installiert ist, müssen Sie ihn manuell installieren. Das Gleiche gilt, wenn der Agent nicht auf die neueste Version aktualisiert wird.
- Time/Load Auto Scaling-Instances — Das Detachment-Tool unterstützt keine Instances mit aktiviertem Auto Scaling. Sie müssen Auto Scaling für Instances deaktivieren, die Sie trennen möchten.
- Berechtigungen — Das Detachment-Tool erstellt oder generiert keine IAM-Entitäten, die auf der Seite „Berechtigungen“ der Konsole angegeben sind. OpsWorks
- Apps — Das Detachment-Tool erstellt oder generiert keine Apps außerhalb von. OpsWorks

Erste Schritte

Schritt 1: Stellen Sie sicher, dass die Voraussetzungen erfüllt sind

Bei allen drei Befehlen des Werkzeugs Detach In Place handelt es sich um Python-Skripts, die Sie lokal, auf einer EC2-Instanz oder mithilfe von AWS CloudShell

AWS CloudShell ist eine browserbasierte Shell, mit der Sie über die Befehlszeile auf die Ressourcen in der AWS ausgewählten Datei zugreifen können. AWS-Region AWS CloudShell ist mit gängigen Tools (wie AWS CLI Python) vorinstalliert. Bei der Verwendung AWS CloudShell verwenden Sie dieselben Anmeldeinformationen, mit denen Sie sich an der Konsole anmelden.

Bei dieser exemplarischen Vorgehensweise wird davon ausgegangen, dass Sie verwenden AWS CloudShell.

Schritt 2: Laden Sie das Skript herunter

1. Laden Sie die ZIP-Datei herunter, die das Migrationsskript und alle relevanten Dateien enthält, indem Sie den folgenden Befehl ausführen:

   ```
   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. Entpacken Sie die Datei, indem Sie den folgenden Befehl ausführen.

   ```
   unzip detach_in_place_script.zip
   ```

   Nach dem Entpacken der Datei sind die folgenden Dateien verfügbar:

   - README.md
   - LIZENZ
   - NOTICE
   - requirements.txt
   - TODO.py

3. Falls erforderlich, installieren Sie, `pipenv` indem Sie den folgenden Befehl ausführen.

   ```
   pip install pipenv
   ```

Schritt 3: Führen Sie das Skript aus

Richten Sie zunächst Ihre Umgebung so ein, dass Sie das Skript ausführen können, indem Sie die folgenden Befehle ausführen.

```
pipenv install -r requirements.txt
pipenv shell
```

Überprüfen Sie dann die Skriptparameter.

Befehl	Parameter	Beschreibung	Typ	Erforderlich	Standard
`handle-prerequisites`	`--layer-id`	Die ID der Ebene, die Sie trennen möchten.	String	Ja	-
	`--region`	Die Region des OpsWorks Stapels. Wenn sich Ihre OpsWorks Stack-Region und Ihre API-Endpunktregion unterscheiden, verwenden Sie die Stack-Region. Dies ist dieselbe Region wie die anderen Ressourcen in Ihrem OpsWorks Stack (z. B. EC2-Instances und Subnetze).	String	Nein	us-east-1
`detach`	`--layer-id`	ID der Ebene, die Sie trennen möchten.	String	Ja	-
	`--batch-size`	Anzahl der Instanzen, die von einer Ebene getrennt werden sollen (z. B. 5).	String	Nein	-
	`--region`	Die Region des OpsWorks Stacks. Wenn sich Ihre OpsWorks Stack-Reg	String	Nein	us-east-1

Befehl	Parameter	Beschreibung	Typ	Erforderlich	Standard
		ion und Ihre API-Endpunktregion unterscheiden, verwenden Sie die Stack-Region. Dies ist dieselbe Region wie die anderen Ressourcen in Ihrem OpsWorks Stack (z. B. EC2-Instances und Subnetze).			
`cleanup`	`--stack-id`	ID des Stacks, den Sie löschen möchten.	String	Nein	Wenn Sie sich gegenseitig ausschließen, müssen Sie entweder eine Layer-ID oder eine Stack-ID angeben
	`--layer-id`	ID der Ebene, die Sie löschen möchten	String	Nein	

Befehl	Parameter	Beschreibung	Typ	Erforderlich	Standard
	`--region`	Die Region des OpsWorks Stapels. Wenn sich Ihre OpsWorks Stack-Region und Ihre API-Endpunktregion unterscheiden, verwenden Sie die Stack-Region. Dies ist dieselbe Region wie die anderen Ressourcen in Ihrem OpsWorks Stack (z. B. EC2-Instances und Subnetze).	String	Nein	us-east-1

Sie können sich die verfügbaren Optionen für die `cleanup` Befehle `handle-prerequisites` und anzeigen lassen`detach`, indem Sie die Befehle mit der `--help` Option wie folgt ausführen:

```
python3 layer_detacher.py detach --help
python3 layer_detacher.py handle-prerequisites --help
python3 layer_detacher.py cleanup --help
```

Sie sind jetzt bereit, loszulegen. Die folgenden Beispiele zeigen, wie Sie die Befehle für verschiedene Anwendungsfälle ausführen können.

Beispiele:

- Beispiel 1: Prüfen Sie, ob ein Layer alle Voraussetzungen erfüllt und abgetrennt werden kann
- Beispiel 2: Trennen Sie alle Instanzen einer Ebene
- Beispiel 3: Trennen Sie alle Instanzen einer Ebene stapelweise
- Beispiel 4: Bereinigen Sie alle Ressourcen für eine Ebene und löschen Sie die Ebene
- Beispiel 5: Bereinigen Sie alle Ressourcen für einen Stack und löschen Sie den Stack

Beispiel 1: Prüfen Sie, ob ein Layer alle Voraussetzungen erfüllt und abgetrennt werden kann

Der folgende Befehl liest Informationen über eine OpsWorks Ebene (und die darin enthaltenen Instanzen) und prüft, ob die folgenden Voraussetzungen erfüllt sind:

- Alle Instanzen sind online.
- Es gibt keine Load/Time Auto Scaling Scaling-Instanzen.
- Alle Instanzen haben den neuesten OpsWorks Agenten.
- Auf allen Instanzen ist der neueste SSM-Agent installiert und konfiguriert.
- Alle Instanzen haben ein SSH-Schlüsselpaar.
- Jede Instanz gehört zu genau einer Ebene.

```
python3 layer_detacher.py handle-prerequisites \
--layer-id opsworks-layer-id \
--region opsworks-stack-region
```

Beispiel 2: Trennen Sie alle Instanzen einer Ebene

Der folgende Befehl iteriert über alle Instanzen des Layers, prüft, ob die Instanzen die Voraussetzungen erfüllen, und versucht, alle Instanzen, die die Voraussetzungen erfüllen, parallel zu trennen. Wenn eine oder mehrere Voraussetzungen nicht erfüllt sind, stellt der Befehl eine Option zur erzwungenen Trennung für die verbleibenden nicht konformen Instanzen bereit.

Vor dem Trennen einer Instanz führt der Befehl wie folgt aus:

1. Speichern Sie das benutzerdefinierte JSON und laden Sie es auf S3 hoch.
2. Erstellen Sie SSM-Automatisierungsdokumente für jedes OpsWorks Lebenszyklusereignis für die Ebene und laden Sie die Ausführungsprotokolle für die Automatisierungsdokumente auf S3 hoch.
3. Erstellen Sie eine AppRegistry Anwendung für alle Instanzen, die getrennt werden sollen. Der Anwendung ist eine Ressourcengruppe zugeordnet, die alle getrennten Instanzen und Ressourcen enthält. Zu den Ressourcen gehören SSM-Automatisierungsdokumente und SSM-Parameter, die Informationen über Lebenszyklusereignisse und benutzerdefinierte Chef-Rezepte enthalten.
4. Trennt den Classic Load Balancer von der Ebene, falls vorhanden.

Mit diesem Befehl werden nur OpsWorks Ressourcen geändert. Der Status der EC2-Instances bleibt unverändert.

```
python3 layer_detacher.py detach \
--layer-id opsworks-layer-id \
--region opsworks-stack-region
```

Beispiel 3: Trennen Sie alle Instanzen einer Ebene stapelweise

Der folgende Befehl macht dasselbe wie das vorherige Beispiel. Der einzige Unterschied besteht darin, dass die Instanzen stapelweise getrennt werden.

Mit diesem Befehl werden nur OpsWorks Ressourcen geändert. Der Status der EC2-Instances bleibt unverändert.

```
python3 layer_detacher.py detach \
--layer-id opsworks-layer-id \
--region opsworks-stack-region \
--batch-size 5
```

Beispiel 4: Bereinigen Sie alle Ressourcen für eine Ebene und löschen Sie die Ebene

Mit dem folgenden Befehl wird über alle Ressourcen einer Ebene iteriert und diese gelöscht. Genauer gesagt werden alle Instances in und EC2 gestoppt und gelöscht, der Load Balancer getrennt OpsWorks und Amazon RDS-Instances, Elastic IPs und Volumes deregistriert. Nach dem Bereinigen der Ressourcen wird die Ebene gelöscht.

Dieser Befehl löscht OpsWorks Ressourcen und EC2-Instances. Wenn Sie möchten, dass Ihre EC2-Instances unangetastet bleiben, verwenden Sie den `detach` Befehl, bevor Sie den Befehl verwenden. `cleanup` Auf diese Weise löscht der `cleanup` Befehl alle verbleibenden Ressourcen.

```
python3 layer_detacher.py cleanup \
--layer-id opsworks-layer-id \
--region opsworks-stack-region
```

Beispiel 5: Bereinigen Sie alle Ressourcen für einen Stack und löschen Sie den Stack

Der folgende Befehl iteriert über alle Ebenen und dann über die Ressourcen jeder Ebene. Für jede Ebene stoppt und löscht der Befehl alle Instances in und EC2, trennt Load Balancer OpsWorks und hebt die Registrierung von Amazon RDS-Instances, Elastic IPs und Volumes auf. Anschließend löscht der Befehl die Ebene. Derselbe Vorgang wird in jeder Ebene ausgeführt, die zu diesem Stapel gehört. Nachdem alle Ebenen gelöscht wurden, wird der Stapel schließlich entfernt.

Dieser Befehl löscht OpsWorks Ressourcen und EC2-Instances. Wenn Sie möchten, dass Ihre EC2-Instances unangetastet bleiben, verwenden Sie den `detach` Befehl, bevor Sie den Befehl verwenden. `cleanup` Auf diese Weise löscht der `cleanup` Befehl alle verbleibenden Ressourcen.

```
python3 layer_detacher.py cleanup \
--stack-id opsworks-stack-id \
--region opsworks-stack-region
```

Schritt 4: Verwenden Sie Ihre Ressourcen weiter, nachdem Sie sich von getrennt haben OpsWorks

Nach der Ausführung des `detach` Befehls erstellt das Tool eine neue AWS Service Catalog AppRegistry Anwendung, die der abgelösten Ebene entspricht. Der Name der Anwendung folgt dem Format*layer-name*`---`*layer-id*. Außerdem wird das `OpsWorksLayerId` Tag hinzugefügt, um die Anwendung eindeutig zu identifizieren, die der abgetrennten Ebene entspricht.

Um dieser Anwendung neue AWS Ressourcen hinzuzufügen (z. B. neue EC2-Instances), können Sie einen der folgenden Schritte ausführen:

1. Kennzeichnen Sie die Ressource mit dem eindeutigen Anwendungs-Tag der AppRegistry Anwendung:

   Tag-Schlüssel: `awsApplication`

   Wert: `arn:aws:resource-groups:`*region*`:`*account-id*`:group/`*application-name*`/`*application-id*`>`
2. Führen Sie den Befehl associate-resource aus.

Zusätzlich wird für jede AppRegistry Anwendung eine Ressourcengruppe erstellt. Die Ressourcengruppe enthält die folgenden Tags.

Tag-Schlüssel	Wert
`EnableAWSServiceCatalogAppRegistry`	`TRUE`
`aws:servicecatalog:applicationName`	*application-name*

Tag-Schlüssel	Wert
`aws:servicecatalog:applicationId`	*application-id*
`aws:servicecatalog:applicationArn`	`arn:aws:servicecatalog:` *region*`:`*account-id*` :/applications/` *application-id*

Aufgaben nach der Trennung ausführen

Die folgende Tabelle enthält Informationen zur Ausführung von Aufgaben nach einer Trennung:

Aufgabe	Beschreibung
Ausführung von Lebenszyklusereignissen	Nachdem Sie den `detach` Befehl ausgeführt haben und die Option ausgewählt haben, erstellt das Skript 5 Automatisierungsdokumente, die den 5 OpsWorks Lebenszyklusereignissen entsprechen. Der Name jedes Automatisierungsdokuments folgt diesem Format:*layer-id*`_`*lifecycle-event*`_automation_document` . Um OpsWorks das Verhalten in Systems Manager zu simulieren, müssen Sie Automatisierungsausführungen manuell auslösen, wenn Sie EC2-Instances bereitstellen, beenden oder Rezepte bereitstellen/entfernen.
Benutzerdefiniertes JSON aktualisieren	Benutzerdefiniertes JSON für den Stack und die Ebene wird in einem S3-Bucket gespeichert, der beim Trennen angegeben wurde, oder alternativ in einem neuen S3-Bucket, der erstellt wird. Die für die JSON-Dateien gespeicherten Dateinamen lauten wie folgt:

Aufgabe	Beschreibung
	• `layercustomjson.json` • `stackcustomjson.json`
Ändern Sie Ihre Ausführungsliste für Lebenszyklusereignisse	Die Ausführungsliste für jedes Lebenszyklusereignis ist im entsprechenden Automatisierungsdokument definiert. Um die Ausführungsliste zu ändern, suchen Sie in der AppRegistry Anwendung nach den Automatisierungsdokumenten und ändern Sie den `RunList` Parameter. Der Vorgang zum Aktualisieren von Rezepten und Kochbüchern ist unverändert`AWS-ApplyChefRecipes` , da der Vorgang, den die Automatisierungsdokumente auslösen, dieselbe Quelle unterstützt wie OpsWorks.
Verwaltung von Auto Healing/Auto Scaling	Wenn Sie eine Instanz trennen, wird der OpsWorks Agent deinstalliert. Ohne den Agenten können fehlerhafte Instances OpsWorks nicht automatisch repariert oder ersetzt werden, und Ihre Flotte kann auch nicht auto skaliert werden. Um Auto Scaling fortzusetzen und ausgefallene Instances zu ersetzen, erstellen Sie eine Amazon EC2 Auto Scaling Scaling-Gruppe. Die Gruppe wird neue Instances starten, um die gewünschte Kapazität aufrechtzuerhalten, wenn Amazon EC2 fehlerhafte Instances erkennt, die ersetzt werden müssen.

Aufgabe	Beschreibung
Load Balancer verwalten	Wenn Ihr Layer einen Classic Load Balancer verwendet, trennt der `detach` Befehl ihn, bevor die Instances deregistriert werden. Auf diese Weise wird sichergestellt, dass alle ELB-Instance-Zuordnungen während der Trennung auf Amazon EC2 erhalten bleiben, sodass Ausfallzeiten vermieden werden. Nach Abschluss des Vorgangs können Sie Ihren ELB auf EC2 verwalten.
Herstellen einer Verbindung zu Ihren Instances	Wenn Sie den `detach` Befehl `handle-prerequisites` oder ausführen, werden zwei Prüfungen durchgeführt: • Die Version des SSM-Agenten und die Berechtigungen • SSH-Schlüssel Die Befehle bieten Ihnen auch die Möglichkeit, den SSM-Agenten zu aktualisieren und die erforderlichen Berechtigungen hinzuzufügen, sodass Sie mithilfe des Sitzungs-Managers eine Verbindung zu Instanzen herstellen können. Wenn SSH-Schlüssel vorhanden sind, haben Sie auch die Möglichkeit, per SSH auf die Instanz zuzugreifen.

Verwenden der Registerkarte „Instanzen“ von Systems Manager Application Manager

Nach dem Trennen können Sie Ihre Instanzen auf der Registerkarte „Application Manager-Instanzen“ anzeigen und verwalten.

Die Registerkarte „Instanzen“ enthält zusammengefasste Informationen über die EC2-Instances einer Anwendung, z. B. ihren Status, ihren Integritätsstatus und den Status des letzten Befehls. Auf dieser Registerkarte können Sie detaillierte Informationen zu einzelnen Instanzen anzeigen, z. B.

den Befehlsverlauf, den Alarmstatus, den Zustand des Systems Manager Manager-Agenten und mehr. Die Registerkarte Instances bietet auch eine Vielzahl von Aktionen, z. B. die Möglichkeit, Chef-Rezepte anzuwenden, eine Instance zu starten oder zu stoppen oder eine Instance zu einer Auto Scaling Scaling-Gruppe hinzuzufügen oder zu entfernen.

Erste Schritte mit AWS OpsWorks Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks bietet eine Vielzahl anpassbarer Komponenten, die Sie kombinieren können, um einen Stack zu erstellen, der Ihren spezifischen Zwecken entspricht. Die Herausforderung für neue Benutzer besteht darin zu verstehen, wie sie diese Komponenten zu einem funktionierenden Stack zusammenstellen und effektiv verwalten können. In dieser Anleitung erhalten Sie eine Einführung in diese Themen.

Wenn Sie …	diese Anleitung durcharbeiten möchten:
Erstellen Sie so schnell wie möglich einen Beispiel-Stack.	Erste Schritte: Beispiel
Experimentieren Sie mit einem Linux-basierten Stack.	Erste Schritte: Linux
Experimentieren Sie mit einem Windows-b asierten Stack.	Erste Schritte: Windows
Lernen Sie, wie Sie Ihre eigenen Chef-Reze ptbücher erstellen.	Erste Schritte: Rezeptbücher

Wenn Sie über vorhandene Rechenressourcen verfügen — Amazon EC2 EC2-Instances oder sogar lokale Instances, die auf Ihrer eigenen Hardware ausgeführt werden — können Sie diese zusammen mit Instances, die Sie mit Stacks erstellt haben, in einen Stack integrieren. AWS OpsWorks Anschließend können Sie AWS OpsWorks Stacks verwenden, um alle zugehörigen Instances als Gruppe zu verwalten, unabhängig davon, wie sie erstellt wurden.

Unterstützung von Regionen

Sie können global auf AWS OpsWorks Stacks zugreifen; Sie können Instanzen auch global erstellen und verwalten. Benutzer können AWS OpsWorks Stacks-Instances so konfigurieren, dass sie in jeder AWS Region außer AWS GovCloud (USA West) und der Region China (Peking) gestartet werden. Um mit AWS OpsWorks Stacks arbeiten zu können, müssen Instances in der Lage sein, eine Verbindung zu einem der folgenden API-Endpunkte für den AWS OpsWorks Stacks-Instanzdienst herzustellen.

Ressourcen können nur in der Region verwaltet werden, in der sie erstellt wurden. Ressourcen, die in einem regionalen Endpunkt erstellt wurden, sind für einen anderen regionalen Endpunkt nicht verfügbar und können auf diesen auch nicht geklont werden. Sie können Instances in beliebigen der folgenden Regionen starten.

- Region USA Ost (Ohio)
- Region USA Ost (Nord-Virginia)
- Region USA West (Oregon)
- Region US West (N. California)
- Region Kanada (Zentral) (nur API, nicht verfügbar für Stacks, die in der erstellt wurden.) AWS Management Console
- Region Asien-Pazifik (Mumbai)
- Region Asien-Pazifik (Singapur)
- Region Asien-Pazifik (Sydney)
- Region Asien-Pazifik (Tokio)
- Region Asien-Pazifik (Seoul)
- Region Europa (Frankfurt)
- Region Europa (Irland)
- Region Europa (London)
- Region Europa (Paris)

- Region Südamerika (São Paulo)

Erste Schritte mit einem Beispiel-Stack

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Diese exemplarische Vorgehensweise zeigt, wie Sie mit AWS OpsWorks Stacks schnell eine Beispielanwendungsumgebung für Node.js mit nur wenigen Mausklicks und ohne das Schreiben von Code erstellen können. Wenn Sie fertig sind, verfügen Sie über eine Amazon Elastic Compute Cloud (Amazon EC2) -Instance, auf der Chef 12 ausgeführt wird, einen Node.js HTTP-Server und eine Web-App, mit der Sie mit Twitter interagieren und Kommentare auf einer Webseite hinterlassen können.

> **Note**
>
> Da beim Abschluss dieser exemplarischen Vorgehensweise automatisch eine Instance mit dem Typ c3.large erstellt wird, können Sie diese exemplarische Vorgehensweise oder das Tool zur Erstellung von Beispielstapeln in AWS OpsWorks Stacks im kostenlosen Kontingent nicht verwenden.AWS Durch die Verwendung des Tools zur Erstellung eines Beispielstapels in einer VPC wird zwar eine t2.medium-Instance erstellt, aber VPCs sind derzeit nicht im kostenlosen Kontingent verfügbar.AWS

Schritt 1: Erfüllen der Voraussetzungen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Um mit der Anleitung beginnen zu können, müssen Sie die folgenden Einrichtungsschritte ausführen. Zu diesen Einrichtungsschritten gehören die Registrierung für ein AWS Konto, die Erstellung eines Administratorbenutzers und die Zuweisung von Zugriffsberechtigungen für Stacks. AWS OpsWorks

Themen

- Registriere dich für ein AWS-Konto
- Erstellen Sie einen Benutzer mit Administratorzugriff
- Weisen Sie Dienstzugriffsberechtigungen zu

Registriere dich für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu https://aws.amazon.com/ auf und klicken Sie auf Mein Konto.

Erstellen Sie einen Benutzer mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen Sie einen Benutzer mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis im AWS IAM Identity Center Benutzerhandbuch.

Melden Sie sich als Benutzer mit Administratorzugriff an

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal.

Weisen Sie weiteren Benutzern Zugriff zu

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

   Anweisungen finden Sie im Benutzerhandbuch unter Einen Berechtigungssatz erstellen.AWS IAM Identity Center
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Anweisungen finden Sie im AWS IAM Identity Center Benutzerhandbuch unter Gruppen hinzufügen.

Weisen Sie Dienstzugriffsberechtigungen zu

Ermöglichen Sie den Zugriff auf den AWS OpsWorks Stacks-Dienst (und die zugehörigen Dienste, auf die AWS OpsWorks Stacks angewiesen ist), indem Sie Ihrer Rolle oder Ihrem `AmazonS3FullAccess` Benutzer die Berechtigungen `AWSOpsWorks_FullAccess` und hinzufügen.

Weitere Informationen zum Hinzufügen von Berechtigungen finden Sie unter Hinzufügen von IAM-Identitätsberechtigungen (Konsole).

Nun haben Sie alle Einrichtungsschritte abgeschlossen und können mit dieser Anleitung beginnen.

Schritt 2: Erstellen eines Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

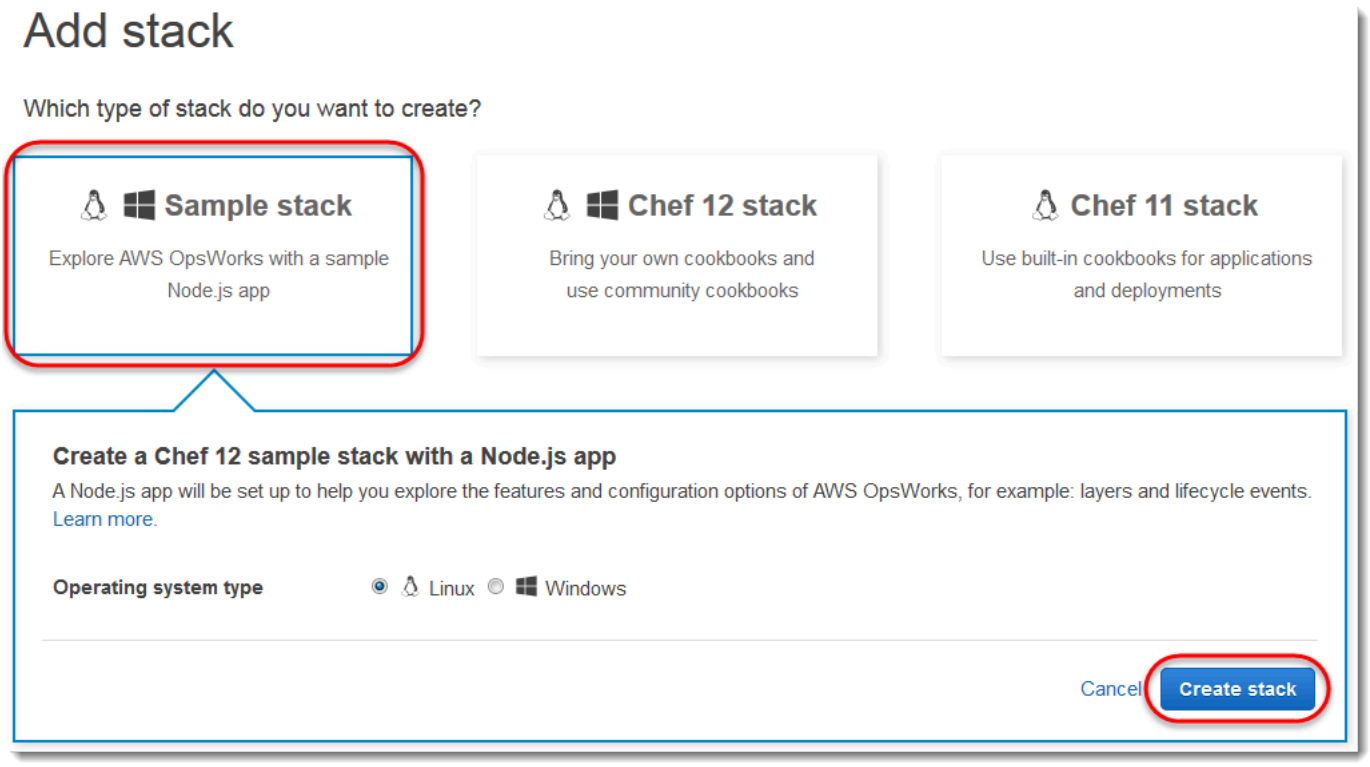
In diesem Schritt verwenden Sie die AWS OpsWorks Stacks-Konsole, um einen Stack zu erstellen. Ein Stack ist eine Sammlung von Instances (wie Amazon EC2 EC2-Instances) und zugehörigen AWS Ressourcen, die einen gemeinsamen Zweck verfolgen und die Sie gemeinsam verwalten

möchten. (Weitere Informationen finden Sie unter Stacks.) Für diese Anleitung wird nur eine Instance verwendet.

Bevor Sie beginnen, müssen Sie die Voraussetzungen erfüllen.

So erstellen Sie den Stack

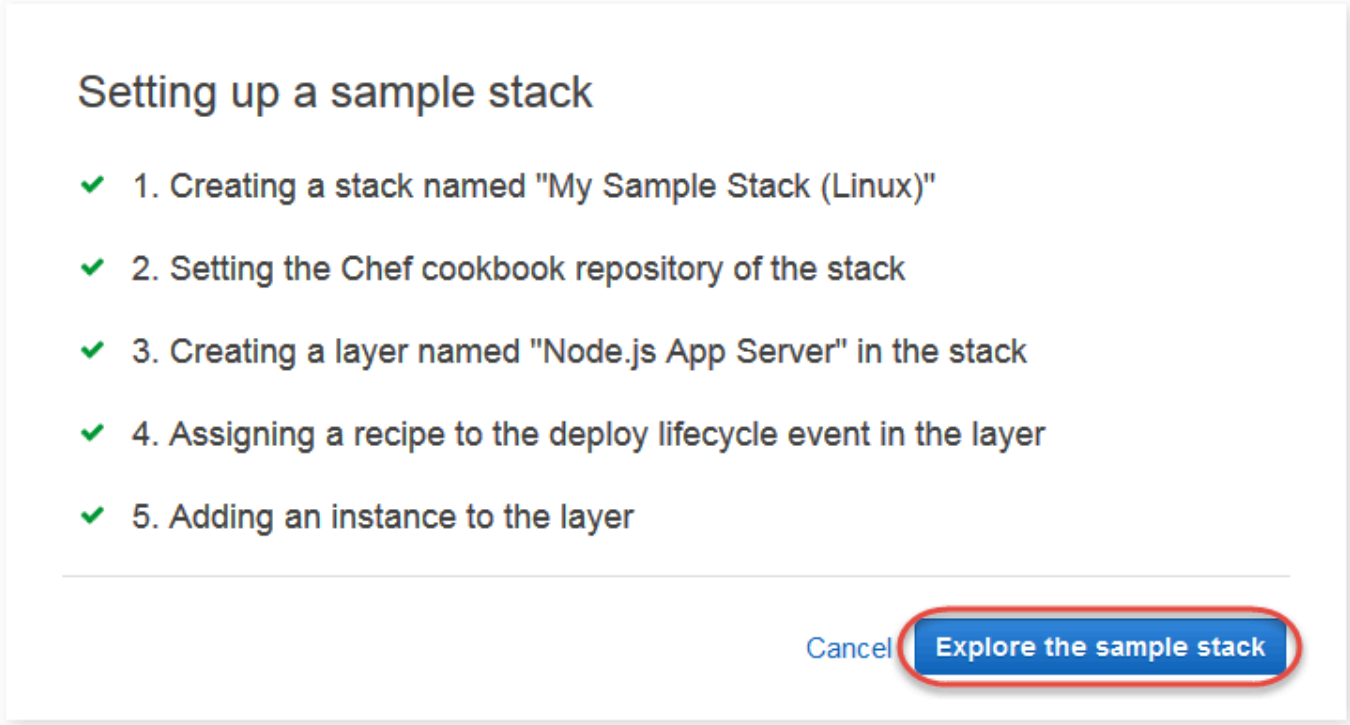
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS OpsWorks Konsole unter https://console.aws.amazon.com/opsworks/.
2. Führen Sie von den folgenden Schritten die zutreffenden aus:
   - Wenn die Seite Willkommen bei AWS OpsWorks Stacks angezeigt wird, wählen Sie Add your first stack oder Add your AWS OpsWorks first Stacks stack (beide Optionen bewirken dasselbe). Die Seite Add stack (Stack hinzufügen) wird angezeigt.
   - Wenn die OpsWorks Dashboard-Seite angezeigt wird, wählen Sie Stapel hinzufügen. Die Seite Add stack (Stack hinzufügen) wird angezeigt.
3. Wählen Sie bei geöffneter Seite Add stack (Stack hinzufügen) die Option Sample stack (Beispiel-Stack) aus, falls sie nicht bereits für Sie ausgewählt ist.
4. Wenn Linux (Linux) bereits als Operating system type (Betriebssystemtyp) ausgewählt ist, wählen Sie Create Stack (Stack erstellen) aus:



5. AWS OpsWorks Stacks erstellt einen Stack mit dem Namen My Sample Stack (Linux). AWS OpsWorks Stacks fügt dem Stack außerdem alle für die Bereitstellung der App erforderlichen Komponenten hinzu:

   - Eine Ebene als Vorlage für eine Gruppe von Instances. Über den Layer werden unter anderem die Einstellungen, Ressourcen, installierten Pakete und Sicherheitsgruppen der Instance festgelegt. (Weitere Informationen finden Sie unter Ebenen.) Die Ebene hat den Namen Node.js App Server (Node.js-Anwendungsserver).
   - Eine Instance, in diesem Fall eine EC2-Instance mit Amazon Linux 2. Weitere Informationen zu Instances finden Sie unter Instances. Der Hostname der Instance ist nodejs-server1 (nodejs-server1).
   - Eine Anwendung, also Code, der auf der Instance ausgeführt wird. Weitere Informationen zu Apps finden Sie unter Apps. Die Anwendung hat den Namen Node.js Sample App (Node.js-Beispielanwendung).

6. Nachdem AWS OpsWorks Stacks den Stack erstellt hat, wählen Sie Explore the sample stack (Linux) aus, um die Seite My Sample Stack (Linux) anzuzeigen (wenn Sie diese exemplarische Vorgehensweise mehrmals durchführen, kann hinter My Sample Stack (Linux) eine fortlaufende Nummer stehen, z. B. 2 oder 3):



Im nächsten Schritt starten Sie die Instance und stellen die Anwendung in der Instance bereit.

Schritt 3: Starten der Instance und Bereitstellen der App

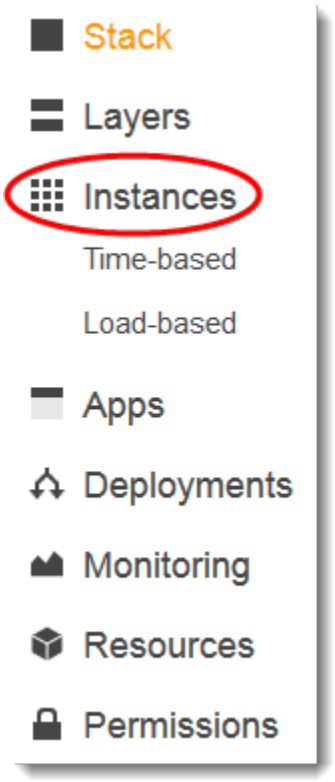
> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Nachdem Sie nun eine Instance und eine App haben, starten Sie die Instance und stellen Sie die App auf der Instance bereit.

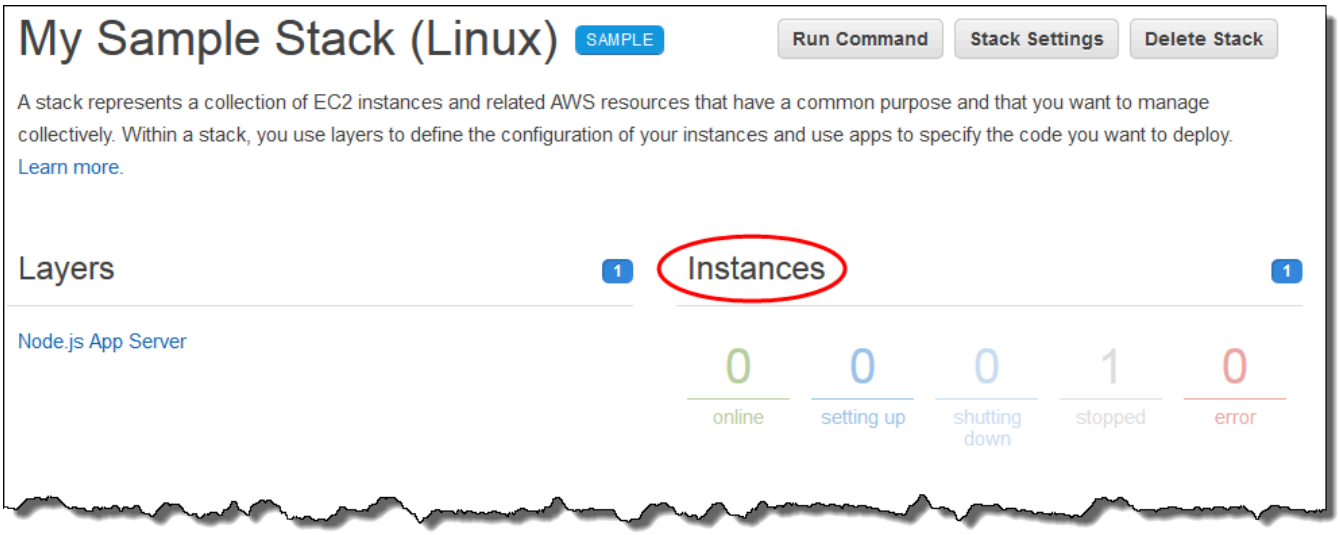
So starten Sie die Instance und stellen die App bereit

1. Führen Sie eine der folgenden Aktionen aus:

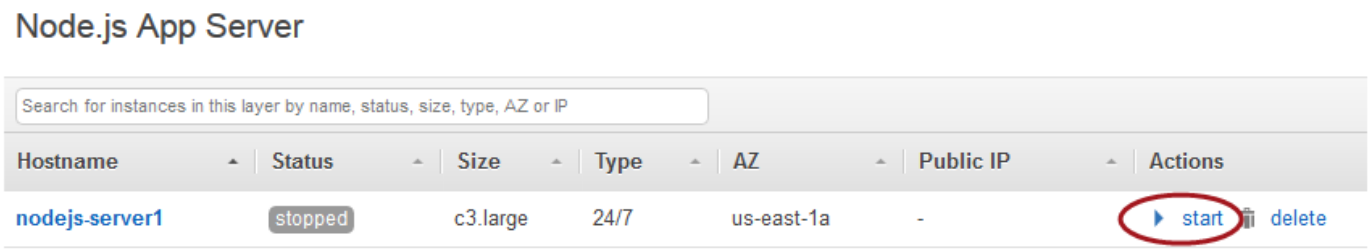
   - Wählen Sie im Service-Navigationsbereich Instances aus:



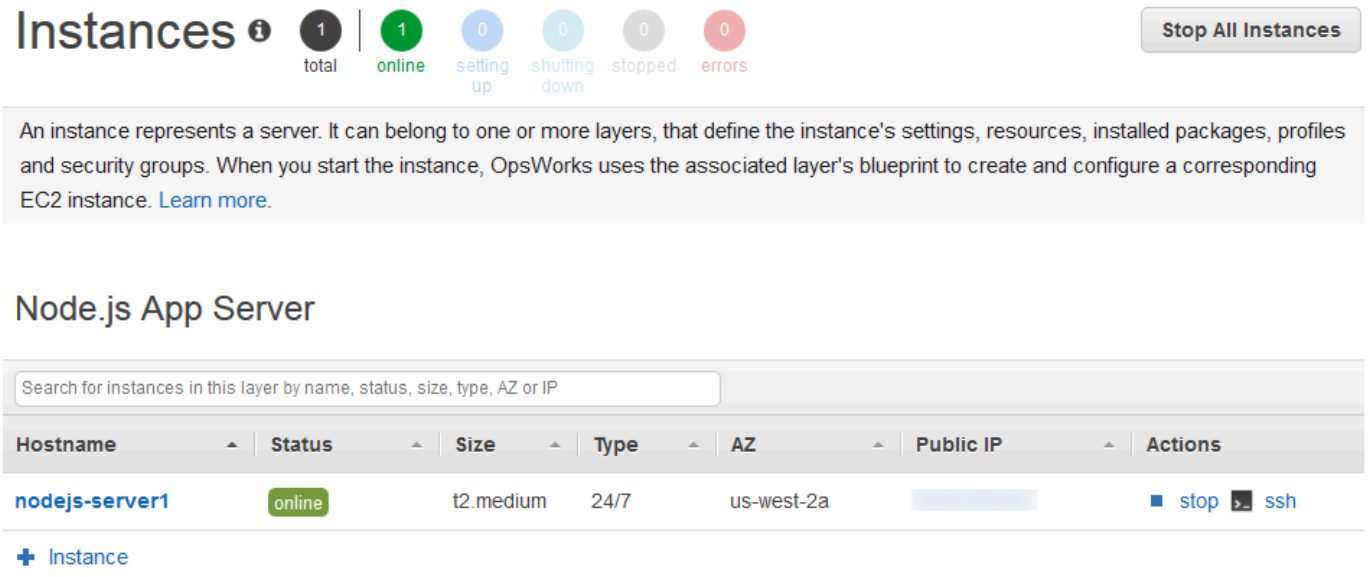
- Wählen Sie auf der Seite My Sample Stack (Linux) (Mein Beispiel-Stack (Linux)) die Option Instances aus:



2. Wählen Sie auf der Seite Instances für Node.js App Server (Node.js-Anwendungsserver) und nodejs server1 (nodejs-server1) die Option start (Starten) aus:



3. Fahren Sie erst fort, wenn der Punkt neben online (Online) grün leuchtet. Wenn eine Fehlermeldung angezeigt wird, lesen Sie unter Handbuch zur Fehlersuche und Fehlerbehebung weiter.
4. Während die Instanz eingerichtet wird, stellt AWS OpsWorks Stacks die App auf der Instanz bereit.
5. Bevor Sie fortfahren, muss Ihr Ergebnis wie auf der Abbildung dargestellt aussehen. Falls eine Fehlermeldung angezeigt wird, lesen Sie unter Handbuch zur Fehlersuche und Fehlerbehebung weiter:



Sie verfügen jetzt über eine Instance mit einer darauf bereitgestellten App.

Im nächsten Schritt werden Sie die Anwendung in der Instance testen.

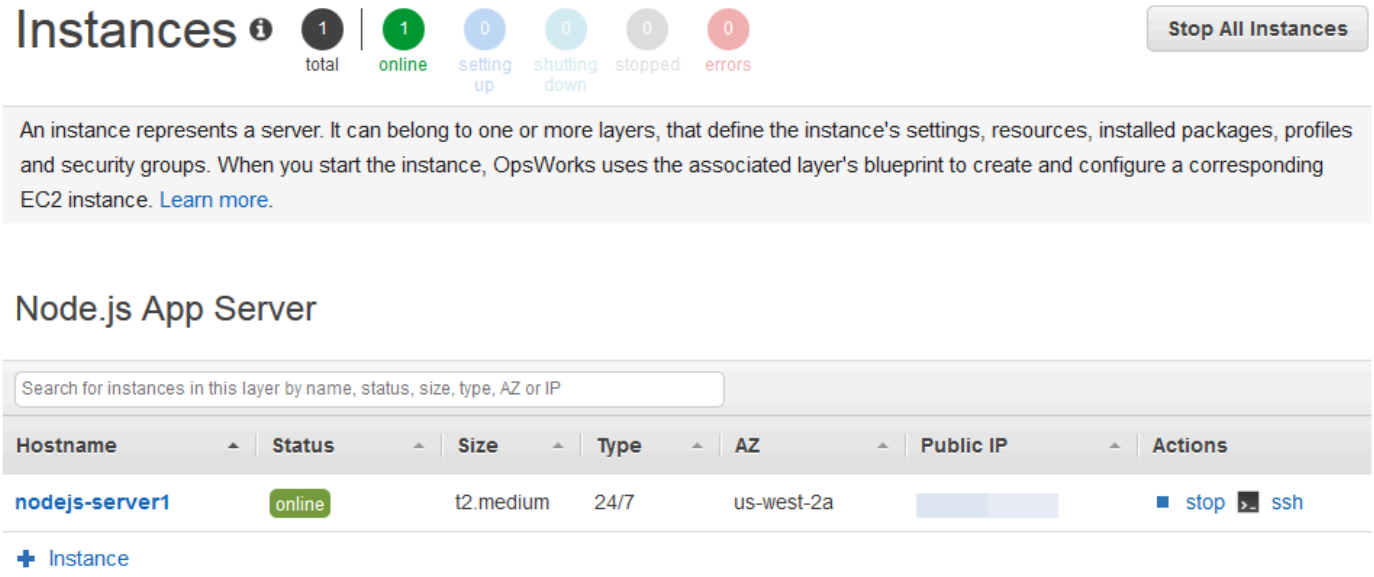
Schritt 4: Testen der bereitgestellten Anwendung in der Instance

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Testen Sie die Ergebnisse der App-Bereitstellung auf der Instance.

So testen Sie die Bereitstellung auf der Instance

1. Wählen Sie auf der im letzten Schritt angezeigten Seite Instances unter Node.js App Server (Node.js-Anwendungsserver), nodejs server1 (nodejs-server1) und Public IP (Öffentliche IP) die IP-Adresse aus.



2. Geben Sie auf der Glückwunsch-Webseite im Textfeld Leave a comment (Kommentar eingeben) einen Kommentar ein und wählen Sie Send (Senden) aus, um die Anwendung zu testen. Die Anwendung fügt den Kommentar zur Webseite hinzu. Sie können beliebig oft Kommentare hinterlassen und Send (Senden) auswählen.

3. Wenn du ein Twitter-Konto hast, wähle Tweet oder Follow @ und folge den Anweisungen auf dem BildschirmAWSOpsWorks, um über die App zu twittern oder @ zu folgen. AWSOpsWorks

Sie haben jetzt die bereitgestellte Anwendung erfolgreich auf der Instance getestet.

In den verbleibenden Schritten können Sie die AWS OpsWorks Stacks-Konsole verwenden, um die Einstellungen des Stacks und seiner Komponenten zu erkunden. Im nächsten Schrittbeginnen Sie dann damit, die Einstellungen des Stacks genauer zu untersuchen.

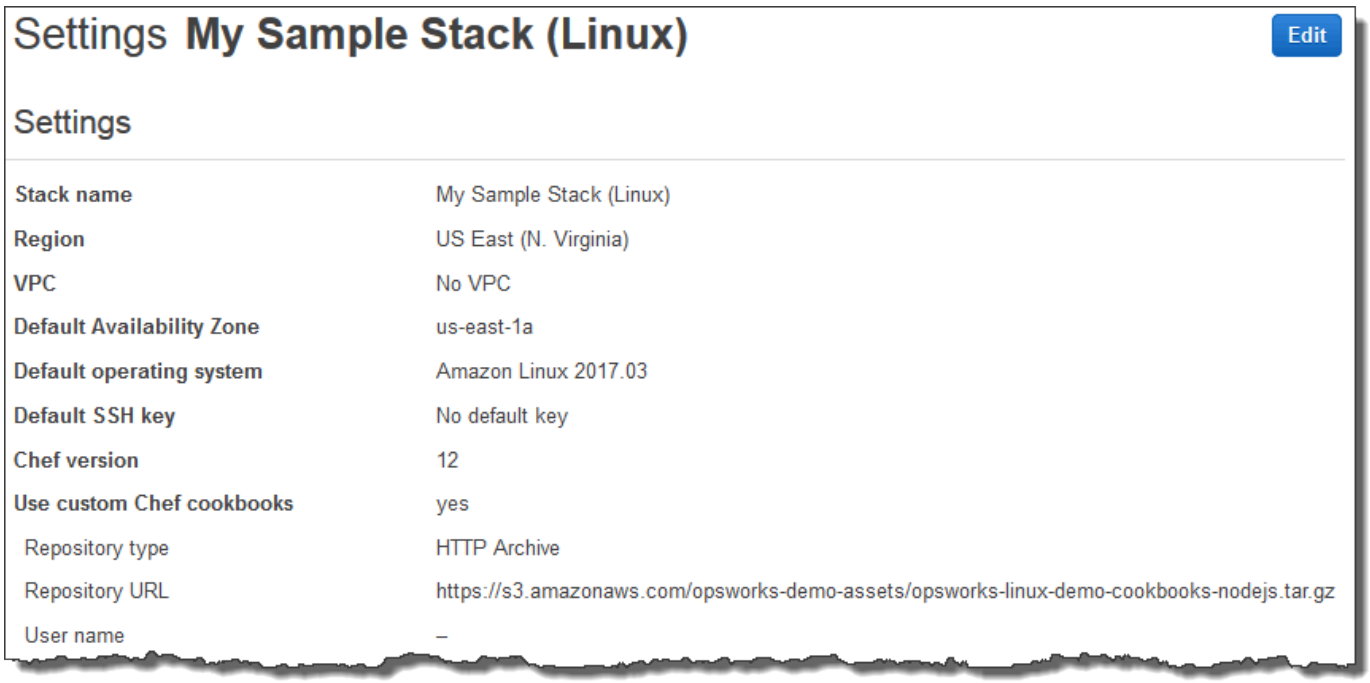
Schritt 5: Betrachten der Stack-Einstellungen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Untersuchen Sie, wie AWS OpsWorks Stacks den Stack eingerichtet hat.

So zeigen Sie die Einstellungen des Stacks an

1. Wählen Sie in der Service-Navigationsleiste Stack (Stack) aus. Die Seite My Sample Stack (Linux) (Mein Beispiel-Stack (Linux)) wird angezeigt.
2. Wählen Sie Stack Settings (Stack-Einstellungen) aus. Die Seite Settings My Sample Stack (Linux) (Einstellungen von Mein Beispiel-Stack (Linux)) wird angezeigt:



Um weitere Informationen zu vielen der Einstellungen zu erhalten, wählen Sie Edit (Bearbeiten) aus und bewegen Sie den Mauszeiger über die einzelnen Einstellungen. Es sind jedoch nicht für alle Einstellungen Beschreibungen verfügbar. Weitere Informationen zu diesen Einstellungen finden Sie unter Erstellen eines neuen Stacks.

Um das in dieser Komplettlösung verwendete Chef-Kochbuch zu erkunden, öffnen Sie das opsworks-linux-demo-cookbooks-nodejs-Repository unter. GitHub

Im nächsten Schritt können Sie die Einstellungen der Ebene genauer untersuchen.

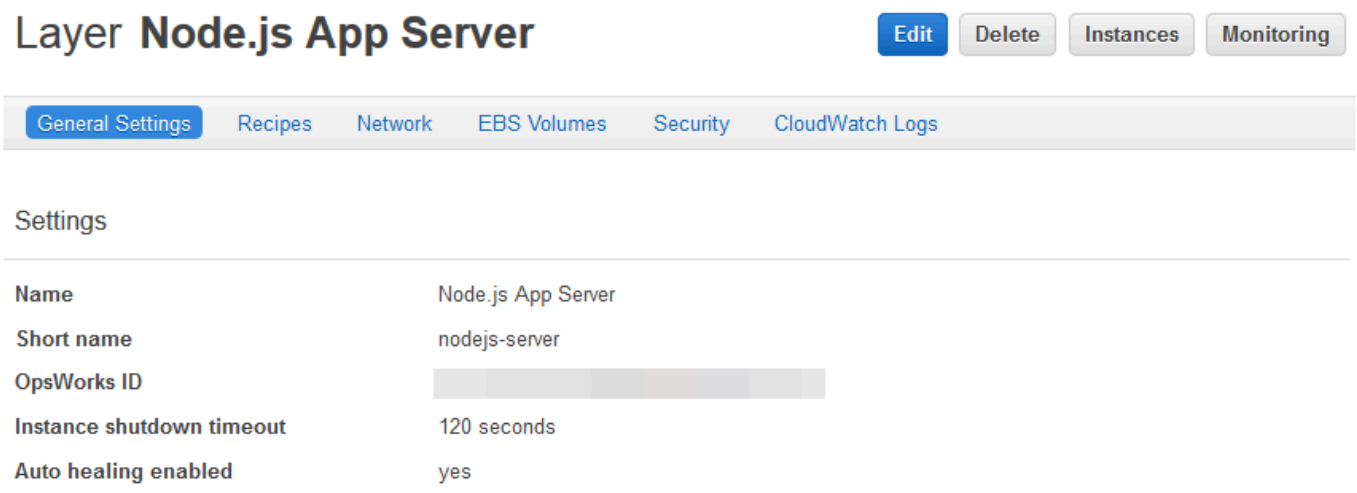
Schritt 6: Betrachten der Einstellungen des Layers

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Untersuchen Sie, wie AWS OpsWorks Stacks die Ebene eingerichtet hat.

So zeigen Sie die Einstellungen des Layers an

1. Wählen Sie im Service-Navigationsbereich Layers aus. Die Seite Layers wird angezeigt.
2. Wählen Sie Node.js App Server (Node.js-Anwendungsserver) aus. Die Seite Layer Node.js App Server (Ebene von Node.js-Anwendungsserver) wird angezeigt. Um die Einstellungen der Ebene anzusehen, wählen Sie General Settings (Allgemeine Einstellungen), Recipes (Rezepte), Network (Netzwerk), EBS Volumes (EBS-Volumes) und Security (Sicherheit) aus.



Um weitere Informationen zu vielen der Einstellungen zu erhalten, wählen Sie Edit (Bearbeiten) aus und bewegen Sie den Mauszeiger über die einzelnen Einstellungen. Es sind jedoch nicht für alle Einstellungen Beschreibungen verfügbar. Weitere Informationen zu diesen Einstellungen finden Sie unter Bearbeiten der Konfiguration einer Ebene OpsWorks .

Im nächsten Schritt können Sie die Einstellungen und Protokolle der Instance genauer betrachten.

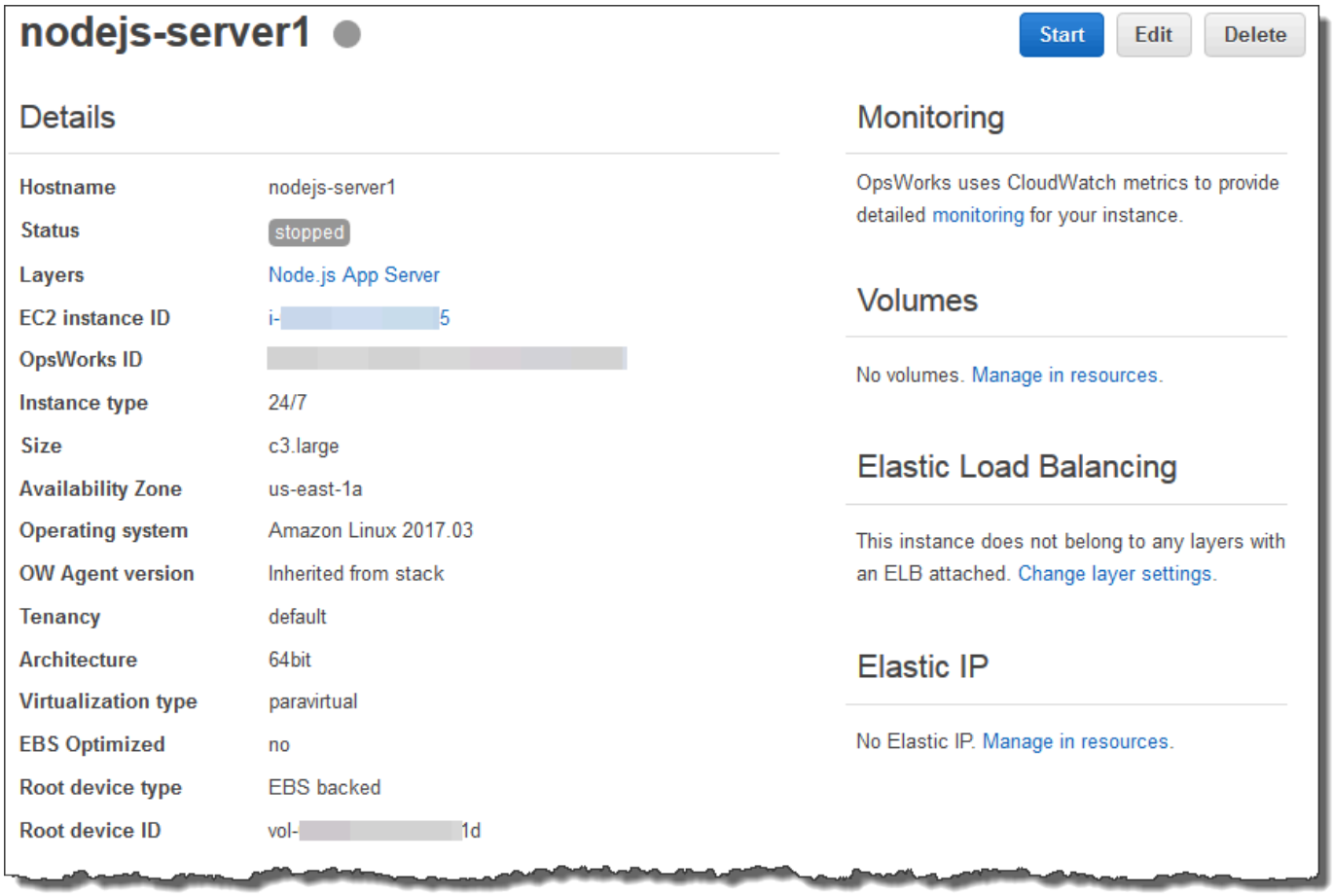
Schritt 7: Betrachten der Einstellungen und Protokolle der Instance

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

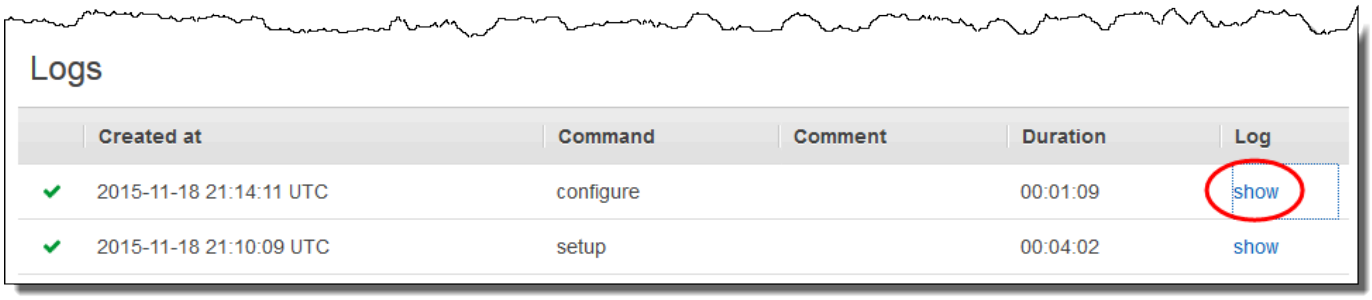
Untersuchen Sie die Einstellungen, die AWS OpsWorks Stacks zum Starten der Instance verwendet hat. Sie können auch die Instanzprotokolle untersuchen, die AWS OpsWorks Stacks erstellt hat.

So zeigen Sie die Einstellungen und Protokolle der Instance an

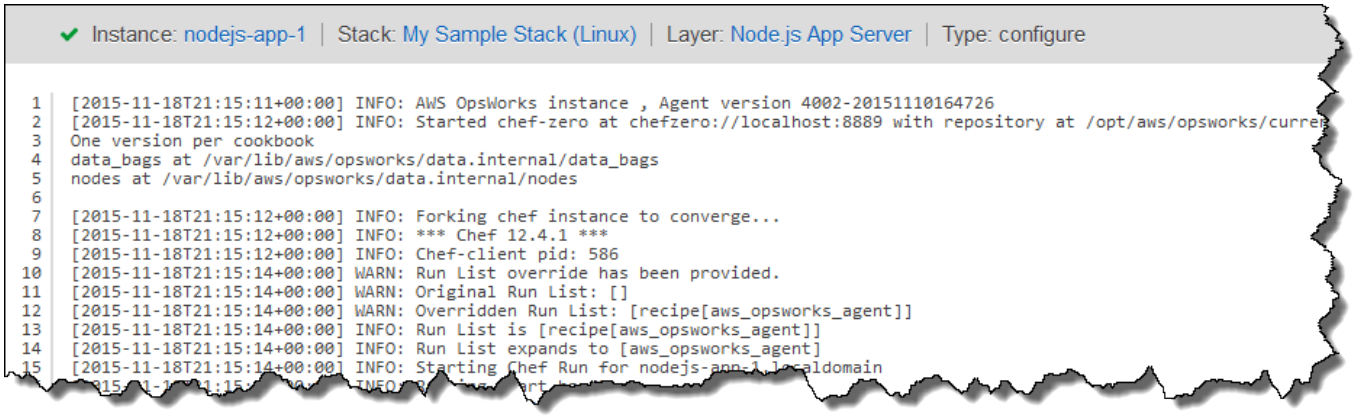
1. Wählen Sie im Service-Navigationsbereich Instances aus. Die Seite Instances wird angezeigt.
2. Wählen Sie für Node.js App Server (Node.js-Anwendungsserver) den Namen nodejs-server1 (nodejs-server1) aus. Die Eigenschaftenseite der Instance wird angezeigt.



3. Um die Protokolle der Instance zu betrachten, wählen Sie im Bereich Logs (Protokolle) für Log (Protokoll) die Option show (Anzeigen) aus.



4. AWS OpsWorks Stacks zeigt das Protokoll in einem separaten Webbrowser-Tab an.



Um weitere Informationen zur Bedeutung einzelner Instance-Einstellungen zu erhalten, rufen Sie die Seite nodejs-server1 (nodejs-server1) erneut auf und wählen Sie Stop (Anhalten) aus. Nachdem die Bestätigung angezeigt wurde, wählen Sie Stop (Anhalten) aus. Wählen Sie Edit (Bearbeiten) aus, nachdem der Status von stopping (Wird angehalten) zu stopped (Angehalten) gewechselt hat, und bewegen Sie den Mauszeiger über die einzelnen Einstellungen. Es sind jedoch nicht für alle Einstellungen Beschreibungen verfügbar. Weitere Informationen zu diesen Einstellungen finden Sie unter Hinzufügen einer Instance zu einem Layer.

Nachdem Sie die Einstellungen überprüft haben, wählen Sie Start (Starten) aus, um die Instance neu zu starten, und warten Sie, bis Status (Status) zu online (Online) wechselt. Solange die Instance angehalten ist, können Sie später die App sonst nicht testen.

> **Note**
>
> Wenn Sie sich bei der Instance anmelden möchten, um sie weiter zu erkunden, müssen Sie AWS OpsWorks Stacks zunächst Informationen über Ihren öffentlichen SSH-Schlüssel

bereitstellen (den Sie mit Tools wie ssh-keygen oder PuTTYgen erstellen können) und dann müssen Sie Berechtigungen für den My Sample Stack (Linux) -Stack festlegen, damit sich Ihr Benutzer bei der Instance anmelden kann. Anweisungen finden Sie unter Registrierung des öffentlichen SSH-Schlüssels eines Benutzers und Anmelden mit SSH.

Im nächsten Schritt lernen Sie die Anwendungseinstellungen kennen.

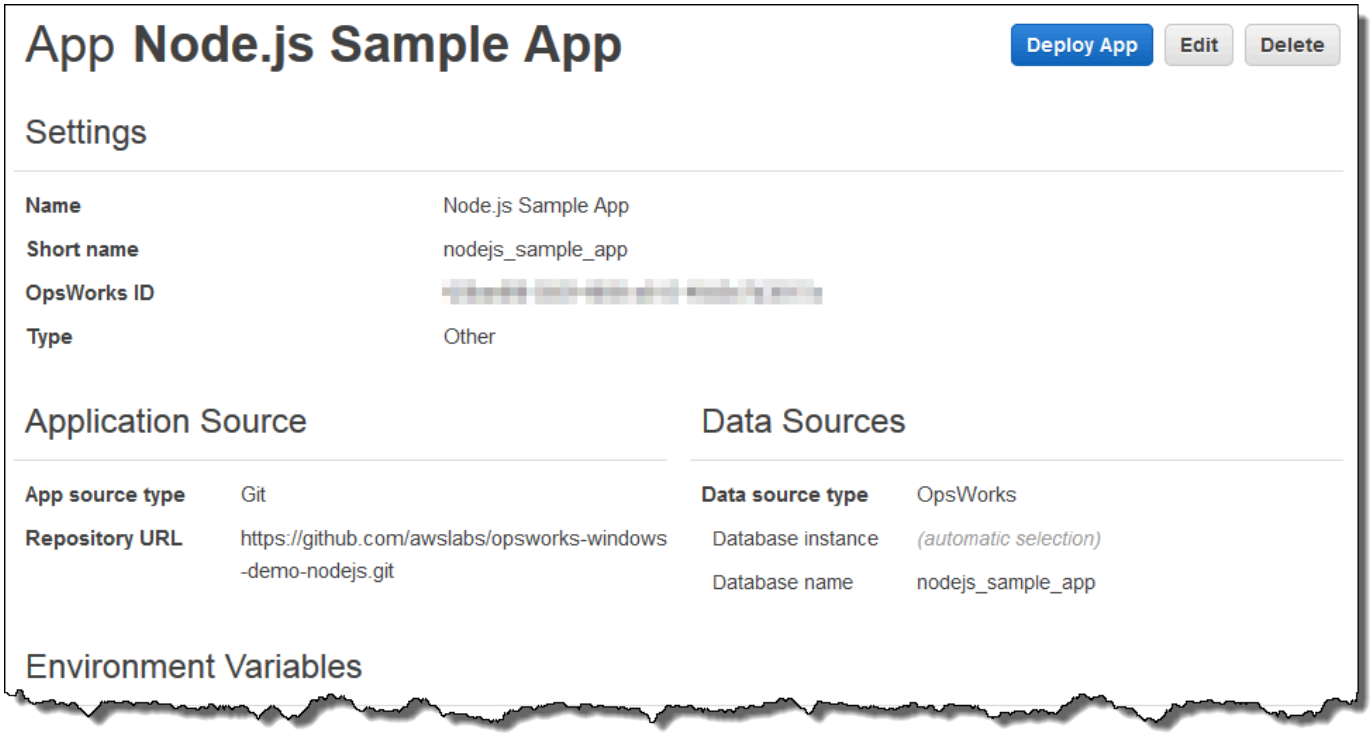
Schritt 8: Betrachten der App-Einstellungen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Untersuchen Sie die Einstellungen, die AWS OpsWorks Stacks für die App verwendet hat.

So zeigen Sie die App-Einstellungen an

1. Wählen Sie im Service-Navigationsbereich Apps (Anwendungen) aus. Die Seite Apps (Anwendungen) wird angezeigt.
2. Wählen Sie Node.js Sample App (Node.js-Beispielanwendung) aus. Die Seite App Node.js Sample App (Anwendung Node.js-Beispielanwendung) wird angezeigt:



Um zu erfahren, was die einzelnen Einstellungen bewirken, wählen Sie Edit (Bearbeiten) aus und bewegen Sie den Mauszeiger über die einzelnen Einstellungen. Es sind jedoch nicht für alle Einstellungen Beschreibungen verfügbar. Weitere Informationen zu den Einstellungen finden Sie unter Hinzufügen von Apps.

Im nächsten Schritt können Sie die Überwachungsberichte für die Ebene betrachten.

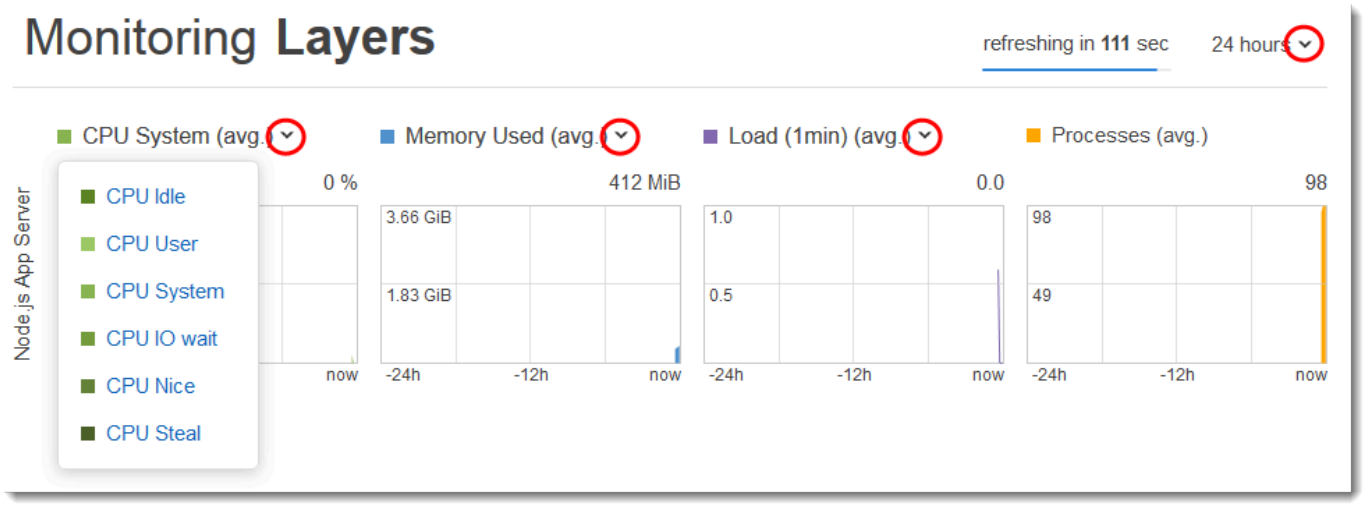
Schritt 9: Betrachten der Layer-Überwachungsberichte

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Untersuchen Sie Berichte, die AWS OpsWorks Stacks über die Rechenleistung der Ebene generiert.

So zeigen Sie die Layer-Überwachungsberichte an

1. Wählen Sie im Service-Navigationsbereich Monitoring (Überwachung) aus. Die Seite Monitoring Layers (Ebenenüberwachung) wird angezeigt.
2. Um weitere Ansichten zu betrachten, wählen Sie die Pfeile neben CPU (CPU), Memory (Arbeitsspeicher), Load (Load) und der Uhrzeit aus:



Weitere Informationen zu diesen und anderen Berichten finden Sie unter Amazon verwenden CloudWatch und Überwachen.

Im nächsten Schritt können Sie zusätzliche Stack-Einstellungen betrachten.

Schritt 10: Betrachten von zusätzlichen Stack-Einstellungen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Schritt können Sie zusätzliche Stack-Einstellungen betrachten.

AWS OpsWorks Stacks führte keine separaten Bereitstellungen durch, stellte keine zusätzlichen Ressourcen bereit und passte keine zusätzlichen Berechtigungen als Teil dieses Stacks an, sodass

die Seiten Bereitstellungen und Befehle, Ressourcen und Berechtigungen nicht sonderlich interessant sind. Wenn Sie sich diese Einstellungen trotzdem ansehen möchten, wählen Sie im Service-Navigationsbereich entsprechend Deployments (Bereitstellungen), Resources (Ressourcen) und Permissions (Berechtigungen) aus. Wenn Sie Informationen zu diesen Seiten benötigen, finden Sie diese unter Bereitstellen von Anwendungen, Ressourcenmanagement und Verwalten von Benutzerberechtigungen.

Im nächsten Schritt können Sie die AWS Ressourcen bereinigen, die Sie für diese exemplarische Vorgehensweise verwendet haben. Dieser nächste Schritt ist optional. Möglicherweise möchten Sie diese AWS Ressourcen weiterhin verwenden, wenn Sie mehr über AWS OpsWorks Stacks erfahren. Wenn Sie diese AWS Ressourcen behalten, kann dies jedoch zu laufenden Gebühren für Ihr AWS Konto führen. Wenn Sie diese AWS Ressourcen für eine spätere Verwendung behalten möchten, haben Sie diese exemplarische Vorgehensweise nun abgeschlossen, und Sie können Nächste Schritte weitermachen.

Schritt 11 (Optional): Bereinigen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Um zu verhindern, dass zusätzliche Gebühren für dein AWS Konto anfallen, kannst du die App und die AWS Ressourcen, die für diese Komplettlösung verwendet wurden, löschen, einschließlich der Instanz und des Stacks-Stacks. AWS OpsWorks (Weitere Informationen finden Sie unter Preisgestaltung.)AWS OpsWorks Möglicherweise möchten Sie diese AWS Ressourcen jedoch weiterhin nutzen, um mehr über AWS OpsWorks Stacks zu erfahren. Wenn Sie diese AWS Ressourcen weiterhin verfügbar halten möchten, haben Sie diese exemplarische Vorgehensweise nun abgeschlossen, und Sie können weitermachen. Nächste Schritte

Inhalte, die in den Ressourcen gespeichert sind, die Sie für diese schrittweise Anleitung erstellt haben, können persönlich identifizierende Informationen enthalten. Wenn Sie nicht mehr möchten, dass diese Informationen von AWS gespeichert werden, führen Sie die in diesem Thema beschriebenen Schritte aus.

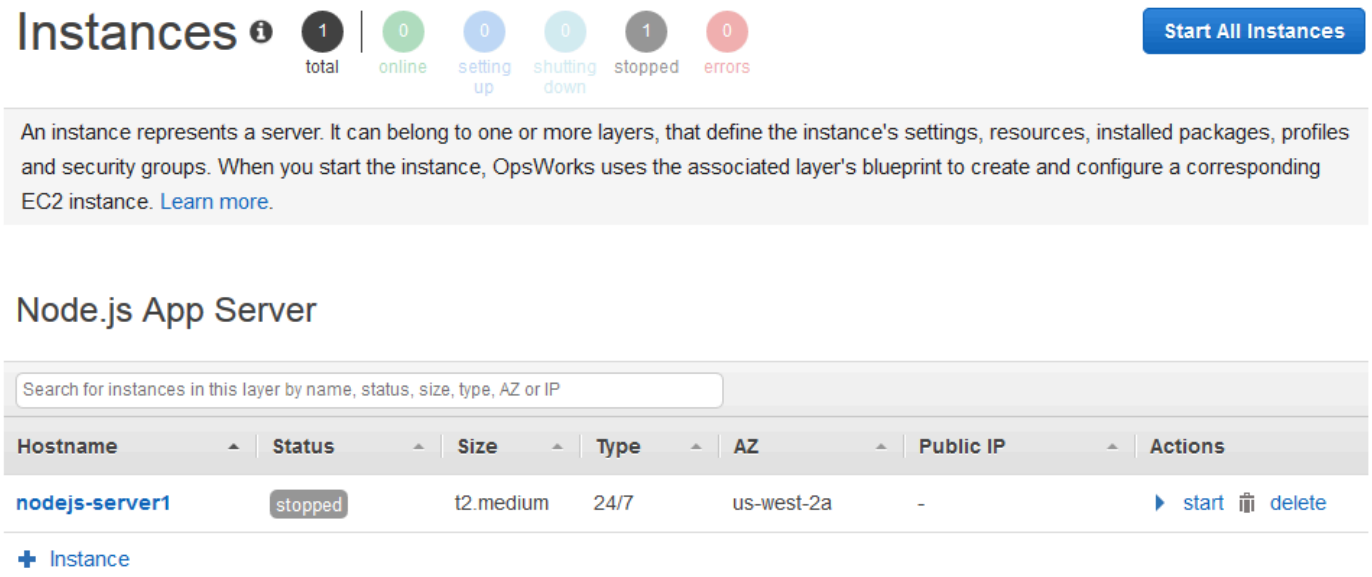
So löschen Sie die Anwendung aus dem Stack

1. Wählen Sie im Service-Navigationsbereich Apps (Anwendungen) aus. Die Seite Apps (Anwendungen) wird angezeigt.
2. Wählen Sie für Node.js Sample App (Node.js-Beispielanwendung) und Actions (Aktionen) die Option delete (Löschen) aus. Wählen Sie im Bestätigungsdialogfeld Delete (Löschen) aus. Nachdem die Anwendung gelöscht wurde, wird die Meldung No apps (Keine Anwendungen) angezeigt.

So löschen Sie die Instance für den Stack

1. Wählen Sie im Service-Navigationsbereich Instances aus. Die Seite Instances wird angezeigt.
2. Wählen Sie für Node.js App Server (Node.js-Anwendungsserver), nodejs-server1 (nodejs-server1) und Actions (Aktionen) die Option stop (Anhalten) aus. Wählen Sie im Bestätigungsdialogfeld Stop aus.

   Dieser Vorgang kann einige Minuten dauern. Wenn AWS OpsWorks Stacks fertig ist, werden die folgenden Ergebnisse angezeigt.



3. Wählen Sie bei Actions (Aktionen) die Option delete (löschen) aus. Wählen Sie im Bestätigungsdialogfeld Delete (Löschen) aus. Die Instance wird gelöscht und die Meldung No instances (Keine Instances) wird angezeigt.

So löschen Sie den Stack

1. Wählen Sie im Service-Navigationsbereich Stack aus. Die Seite My Sample Stack (Linux) (Mein Beispiel-Stack (Linux)) wird angezeigt.
2. Wählen Sie Delete Stack. Wählen Sie im Bestätigungsdialogfeld Delete (Löschen) aus. Der Stapel wird gelöscht und die OpsWorksDashboard-Seite wird angezeigt.

Optional können Sie das Benutzer- und Amazon EC2 EC2-Schlüsselpaar, das Sie für diese exemplarische Vorgehensweise verwendet haben, löschen, wenn Sie sie nicht für den Zugriff auf andere AWS Services und EC2-Instances wiederverwenden möchten. Anweisungen finden Sie unter Löschen eines IAM-Benutzers und Amazon EC2 EC2-Schlüsselpaare und Linux-Instances.

Sie haben diese Anleitung nun abgeschlossen. Weitere Informationen finden Sie unter Nächste Schritte.

Nächste Schritte

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Nachdem Sie diese exemplarische Vorgehensweise abgeschlossen haben, können Sie mehr über die Verwendung von Stacks erfahren: AWS OpsWorks

- Üben Sie, diesen Stapel mithilfe von Stacks manuell neu zu erstellen. AWS OpsWorks Siehe Erste Schritte: Linux.
- Erkunden Sie das Kochbuch und die App, die AWS OpsWorks Stacks für diese Komplettlösung verwendet hat. Weitere Informationen erhalten Sie auch unter Weiterführende Informationen: Arbeiten mit dem Rezeptbuch, das in dieser Anleitung verwendet wird und Weiterführende Informationen: Arbeiten mit der Anwendung, die in dieser Anleitung verwendet wird in der begleitenden Anleitung Erste Schritte: Linux.
- Üben Sie die Verwendung von AWS OpsWorks Stacks mit Windows-Instanzen. Siehe Erste Schritte: Windows.

- Weitere Informationen zu Stacks finden Sie auch unter Erstellen eines neuen Stacks.
- Weitere Informationen zu Layern finden Sie unter Bearbeiten der Konfiguration einer Ebene OpsWorks .
- Weitere Informationen zu Instances finden Sie unter Hinzufügen einer Instance zu einem Layer.
- Weitere Informationen zu Apps finden Sie unter Bereitstellen von Anwendungen.
- Weitere Informationen zu Cookbooks und Rezepte.
- Erstellen Sie Ihre eigenen Rezeptbücher. Siehe Erste Schritte: Rezeptbücher.
- Weitere Informationen zur Zugriffssteuerung für Stacks finden Sie unter Sicherheit und Berechtigungen.

Erste Schritte mit Linux-Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In dieser exemplarischen Vorgehensweise erfahren Sie, wie Sie AWS OpsWorks Stacks verwenden, um eine Node.js Anwendungsumgebung zu erstellen. Wenn Sie fertig sind, verfügen Sie über eine Amazon Elastic Compute Cloud (Amazon EC2) -Instance, auf der Chef 12 ausgeführt wird, einen Node.js HTTP-Server und eine Web-App, mit der Sie mit Twitter interagieren und Kommentare auf einer Webseite hinterlassen können.

Chef ist ein Drittanbieter-Framework zum Konfigurieren und Verwalten von Servern, z. B. EC2-Instances, und zur Bereitstellung und Wartung von Anwendungen auf diesen Servern. Wenn Sie mit Chef nicht vertraut sind, empfehlen wir Ihnen, nach Abschluss dieser exemplarischen Vorgehensweise mehr über Chef zu erfahren, damit Sie alle Vorteile, die AWS OpsWorks Stacks zu bieten hat, voll ausschöpfen können. (Weitere Informationen finden Sie auf der Website Learn Chef.)

AWS OpsWorks Stacks unterstützt vier Linux-Distributionen: Amazon Linux, Ubuntu Server, CentOS und Red Hat Enterprise Linux. Für diese exemplarische Vorgehensweise verwenden wir Ubuntu Server. AWS OpsWorks Stacks funktioniert auch mit Windows Server. Wir haben zwar eine

entsprechende exemplarische Vorgehensweise für Windows Server-Stacks, wir empfehlen jedoch, dass Sie zuerst diese exemplarische Vorgehensweise durchführen, um grundlegende Konzepte zu AWS OpsWorks Stacks und Chef zu erlernen, die dort nicht wiederholt werden. Nachdem Sie diese Anleitung durchgearbeitet haben, rufen Sie die Anleitung Erste Schritte: Windows auf.

Themen

Schritt 1: Erfüllen der Voraussetzungen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Um mit der Anleitung beginnen zu können, müssen Sie die folgenden Einrichtungsschritte ausführen. Zu diesen Einrichtungsschritten gehören die Registrierung für ein AWS Konto, die Erstellung eines Administratorbenutzers und die Zuweisung von Zugriffsberechtigungen für Stacks. AWS OpsWorks

Wenn Sie bereits die Anleitung Erste Schritte: Beispiel durchgearbeitet haben, erfüllen Sie bereits die Voraussetzungen für diese Anleitung und können direkt mit Schritt 2: Erstellen eines Stacks fortfahren.

Themen

- Registriere dich für ein AWS-Konto
- Erstellen Sie einen Benutzer mit Administratorzugriff
- Weisen Sie Dienstzugriffsberechtigungen zu

Registriere dich für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu https://aws.amazon.com/ auf und klicken Sie auf Mein Konto.

Erstellen Sie einen Benutzer mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen Sie einen Benutzer mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis im AWS IAM Identity Center Benutzerhandbuch.

Melden Sie sich als Benutzer mit Administratorzugriff an

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal.

Weisen Sie weiteren Benutzern Zugriff zu

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

Anweisungen finden Sie im Benutzerhandbuch unter Einen Berechtigungssatz erstellen.AWS IAM Identity Center

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Anweisungen finden Sie im AWS IAM Identity Center Benutzerhandbuch unter Gruppen hinzufügen.

Weisen Sie Dienstzugriffsberechtigungen zu

Ermöglichen Sie den Zugriff auf den AWS OpsWorks Stacks-Dienst (und die zugehörigen Dienste, auf die AWS OpsWorks Stacks angewiesen ist), indem Sie Ihrer Rolle oder Ihrem `AmazonS3FullAccess` Benutzer die Berechtigungen `AWSOpsWorks_FullAccess` und hinzufügen.

Weitere Informationen zum Hinzufügen von Berechtigungen finden Sie unter Hinzufügen von IAM-Identitätsberechtigungen (Konsole).

Nun haben Sie alle Einrichtungsschritte abgeschlossen und können mit dieser Anleitung beginnen.

Schritt 2: Erstellen eines Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie werden die AWS OpsWorks Stacks-Konsole verwenden, um einen Stack zu erstellen. Ein Stack ist eine Sammlung von Instanzen und zugehörigen AWS Ressourcen, die einem gemeinsamen Zweck dienen und die Sie gemeinsam verwalten möchten. (Weitere Informationen finden Sie unter Stacks.) Für diese Anleitung gibt es nur eine Instance.

Erfüllen Sie, bevor Sie beginnen, die Voraussetzungen, falls noch nicht geschehen.

So erstellen Sie den Stack

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS OpsWorks Konsole unter https://console.aws.amazon.com/opsworks/.
2. Führen Sie von den folgenden Schritten die zutreffenden aus:
   - Wenn die Seite Willkommen bei AWS OpsWorks Stacks angezeigt wird, wählen Sie Add your first stack oder Add your AWS OpsWorks first Stacks stack (beide Optionen bewirken dasselbe). Die Seite Add stack (Stack hinzufügen) wird angezeigt.
   - Wenn die OpsWorks Dashboard-Seite angezeigt wird, wählen Sie Stapel hinzufügen. Die Seite Add stack (Stack hinzufügen) wird angezeigt.
3. Wählen Sie bei geöffneter Seite Add stack (Stack hinzufügen) die Option Chef 12 stack (Chef 12-Stack) aus, falls sie nicht bereits für Sie ausgewählt ist.
4. Geben Sie in das Feld Stack name (Stack-Name) einen Namen ein, z. B. **`MyLinuxDemoStack`**. (Sie können auch einen anderen Namen eingeben. Stellen Sie jedoch sicher, diesen anstelle von `MyLinuxDemoStack` zu nutzen.)
5. Wählen Sie für Region die Option US West (Oregon) aus.
6. Führen Sie für VPC einen der folgenden Schritte aus:
   - Wählen Sie eine VPC aus, falls verfügbar. (Weitere Informationen finden Sie unter Ausführen eines Stacks in einer VPC.)
   - Wählen Sie andernfalls No VPC (Keine VPC) aus.
7. Wählen Sie für Default operating system (Standardbetriebssystem) die Optionen Linux und Ubuntu 18.04 LTS aus.
8. Bestätigen Sie die Option Use custom Chef cookbooks (Benutzerdefinierte Chef-Rezeptbücher verwenden) mit Yes.
9. Wählen Sie für Repository type (Repository-Typ) die Option Http Archive (HTTP-Archiv) aus.
10. Geben Sie als Repository URL (Repository-URL) die URL **`https://s3.amazonaws.com/opsworks-demo-assets/opsworks-linux-demo-cookbooks-nodejs.tar.gz`**
11. Übernehmen Sie die Standardeinstellungen für die folgenden Schritte:
    - Default Availability Zone (us-west-2a)
    - Default SSH key (Do not use a default SSH key)
    - User name (Benutzername) (leer)
    - Password (Passwort) (leer)

- Stack color (dark blue)

12. Wählen Sie Advanced (Erweitert) aus.
13. Führen Sie für die IAM-Rolle einen der folgenden Schritte aus (weitere Informationen finden Sie unterAWS OpsWorks Stacks erlauben, in Ihrem Namen zu handeln):

    - Wählen Sie aws-opsworks-service-role aus, sofern verfügbar.
    - Falls aws-opsworks-service-rolenicht verfügbar, wählen Sie Neue IAM-Rolle aus.

14. Führen Sie für das Standard-IAM-Instanzprofil einen der folgenden Schritte aus (weitere Informationen finden Sie unterFestlegen von Berechtigungen für Apps auf EC2-Instances):

    - Wenn aws-opsworks-ec2 Rollen verfügbar sind, wählen Sie sie aus.
    - Wenn aws-opsworks-ec2 Rollen nicht verfügbar sind, wählen Sie Neues IAM-Instanzprofil aus.

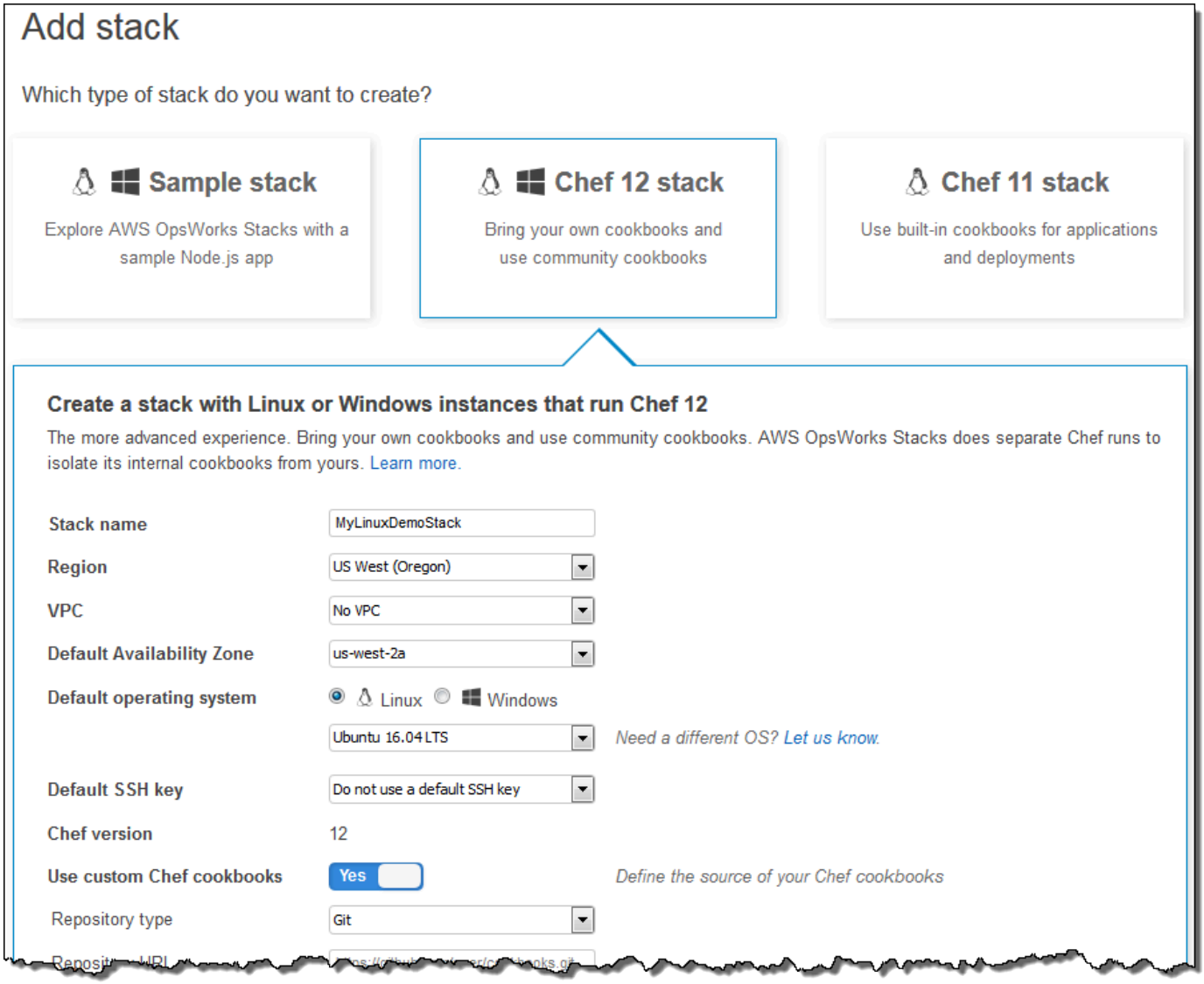
15. Wählen Sie für API endpoint region (API-Endpunktregion) den regionalen API-Endpunkt aus, mit dem der Stack verknüpft sein soll. Wenn sich der Stack in der Region USA West (Oregon) innerhalb des regionalen Endpunkts USA Ost (Nord-Virginia) befinden soll, wählen Sie us-east-1. Wenn der Stack sowohl in der Region USA West (Oregon) als auch mit dem regionalen Endpunkt USA West (Oregon) verknüpft sein soll, wählen Sie us-west-2.

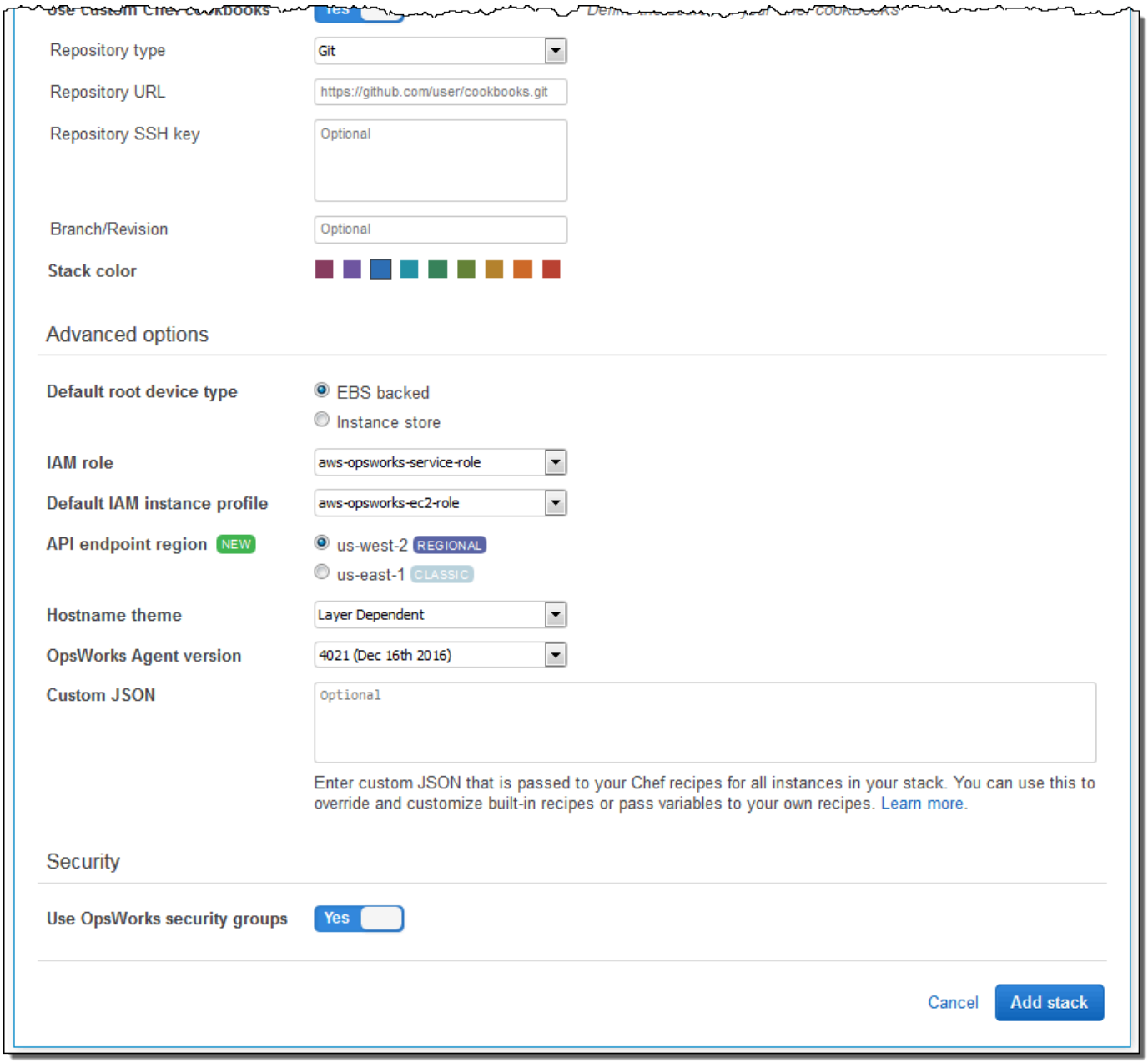
    > **Note**
    >
    > Der regionale Endpunkt USA Ost (Nord-Virginia) enthält aus AWS-Regionen Gründen der Abwärtskompatibilität den älteren Endpunkt. Es hat sich jedoch bewährt, den regionalen Endpunkt zu wählen, der dem Standort, an dem Sie verwalten, am nächsten liegt. AWS Weitere Informationen finden Sie unter Unterstützung von Regionen.

16. Übernehmen Sie die Standardeinstellungen für die folgenden Schritte:

    - Default root device type (EBS backed)
    - Hostname theme (Layer Dependent)
    - OpsWorks Agentenversion (neueste Version)
    - Custom JSON (Benutzerdefinierte JSON-Datei) (leer)
    - Verwenden Sie OpsWorks Sicherheitsgruppen (Ja)

17. Ihre Ergebnisse sollten mit den folgenden Screenshots übereinstimmen, mit Ausnahme von VPC, IAM-Rolle und Standard-IAM-Instanzprofil:


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. Wählen Sie „Stack hinzufügen“. AWS OpsWorks Stacks erstellt den Stapel und zeigt die MyLinuxDemoStackSeite an.

Sie haben nun einen Stack mit den richtigen Einstellungen für diese Anleitung.

Im nächsten Schritt fügen Sie dem Stack eine Ebene hinzu.

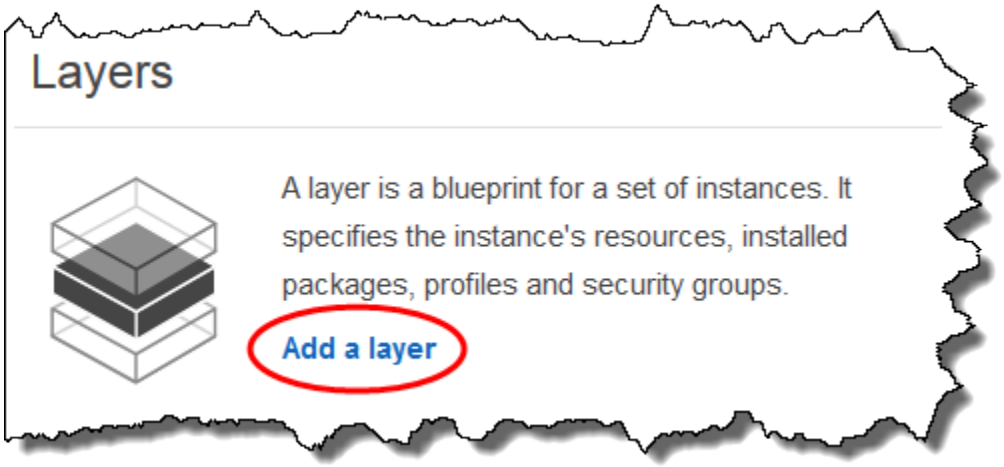
Schritt 3: Hinzufügen eines Layers zum Stack

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Eine Ebene ist ein Blueprint für eine Reihe von Instances, z. B. Amazon EC2 EC2-Instances. Er legt Informationen fest, wie die Instance-Einstellungen, Ressourcen, installierte Pakete und Sicherheitsgruppen. Fügen Sie dem Stack als Nächstes einen Layer hinzu. (Weitere Informationen über Layer finden Sie unter Ebenen.)

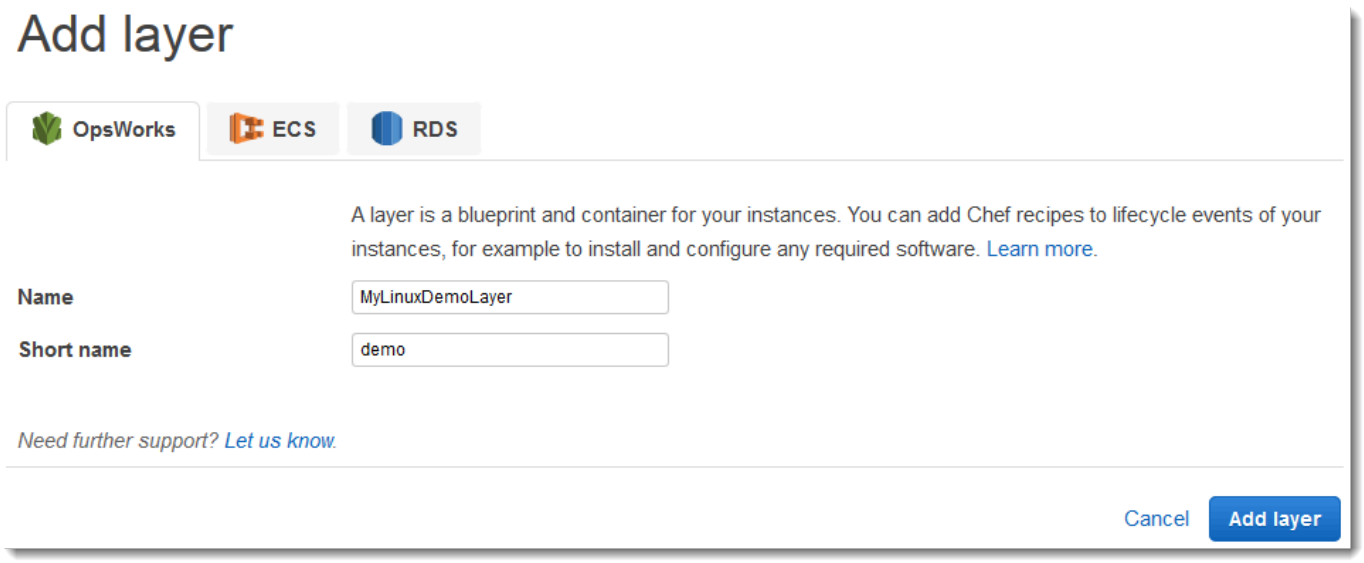
So fügen Sie den Layer dem Stack hinzu

1. Wenn die MyLinuxDemoStackSeite aus dem vorherigen Schritt angezeigt wird, wählen Sie für Ebenen die Option Ebene hinzufügen aus:

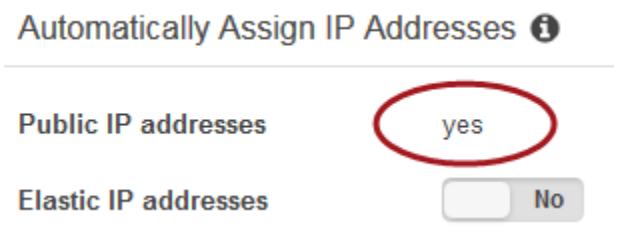


2. Die Seite Add Layer (Ebene hinzufügen) wird angezeigt. Geben Sie auf der OpsWorksRegisterkarte als Name den Text ein**`MyLinuxDemoLayer`**. (Sie können auch einen anderen Namen eingeben. Stellen Sie jedoch sicher, diesen anstelle von `MyLinuxDemoLayer` zu nutzen.)

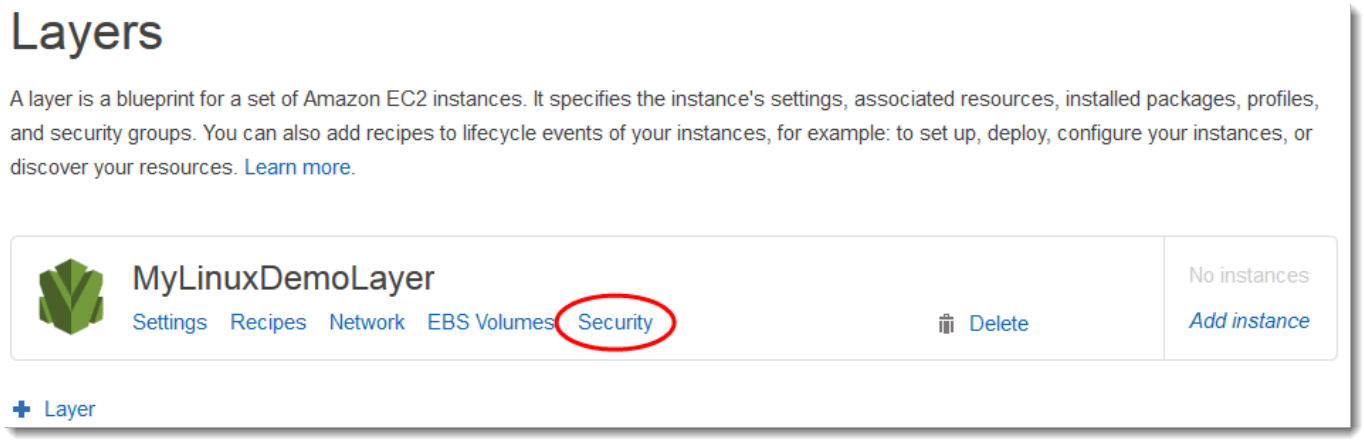
3. Geben Sie für Short name (Kurzname) den Namen **`demo`** ein (Sie können einen anderen Wert eingeben, sollten aber sicherstellen, diesen in der gesamten Anleitung anstelle von `demo` zu nutzen):



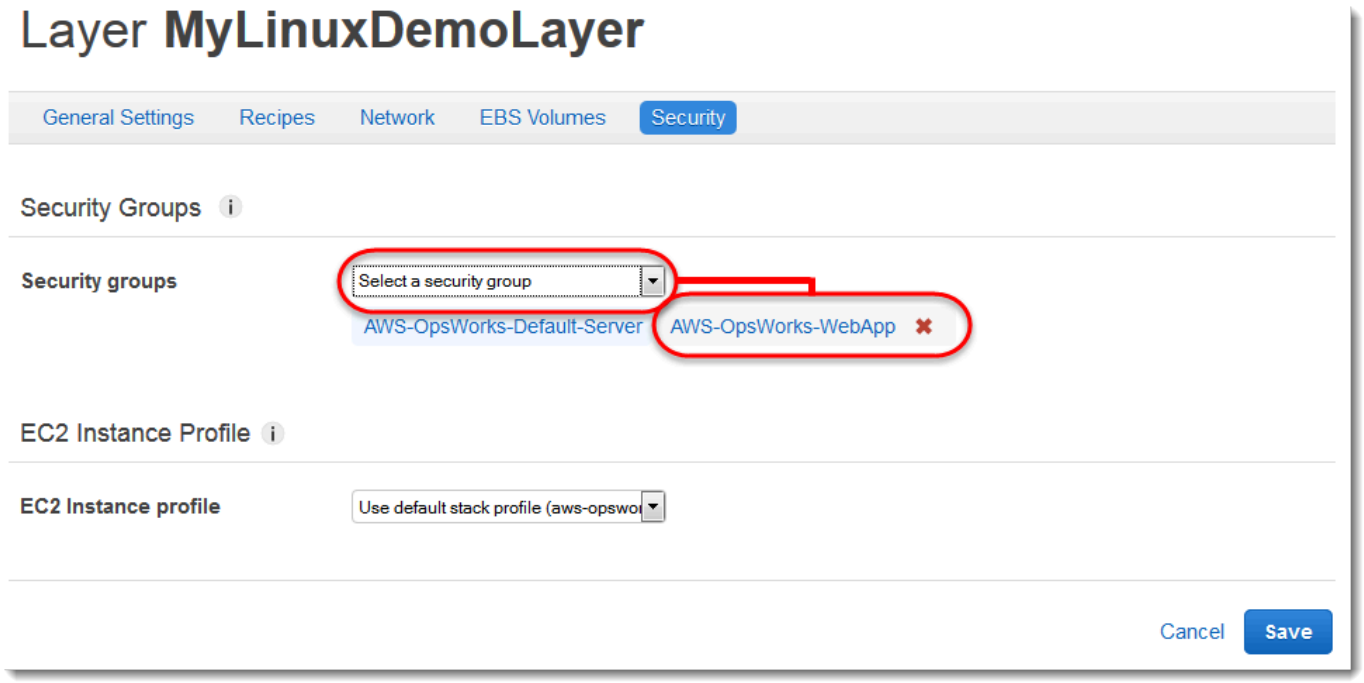
4. Wählen Sie Ebene hinzufügen aus. AWS OpsWorks Stacks erstellt die Ebene und zeigt die Seite „Ebenen“ an.
5. Wählen Sie auf der Seite „Ebenen“ für die MyLinuxDemoLayerOption Netzwerk aus.
6. Überprüfen Sie auf der Registerkarte Network (Netzwerk) unter Automatically Assign IP Addresses (IP-Adressen automatisch zuweisen), ob Public IP addresses (Öffentliche IP-Adressen) auf yes (Ja) festgelegt ist. Wenn Sie Änderungen vorgenommen haben, wählen Sie Save (Speichern) aus.



7. Wählen Sie auf der Seite Layers (Ebenen) die Option Security (Sicherheit) aus:



8. Die MyLinuxDemoLayer Seite „Layer“ wird mit geöffneter Registerkarte „Sicherheit“ angezeigt. Wählen Sie für Sicherheitsgruppen AWS WebApp - OpsWorks - und dann Speichern aus:



9. Die `AWS-OpsWorks-WebApp`-Sicherheitsgruppe wird dem Layer hinzugefügt. (Diese Sicherheitsgruppe ermöglicht es Benutzern, später in dieser exemplarischen Vorgehensweise eine Verbindung mit der App auf der Instance herzustellen. Ohne diese Sicherheitsgruppe erhalten Benutzer in ihrem Webbrowser eine Meldung, dass sie keine Verbindung mit der Instanz herstellen können.)

Sie haben nun einen Layer mit den richtigen Einstellungen für diese Anleitung.

Im nächsten Schritt legen Sie die Anwendung fest, die für die Instance bereitgestellt werden soll.

Schritt 4: Angeben der Anwendung zum Bereitstellen der Instance

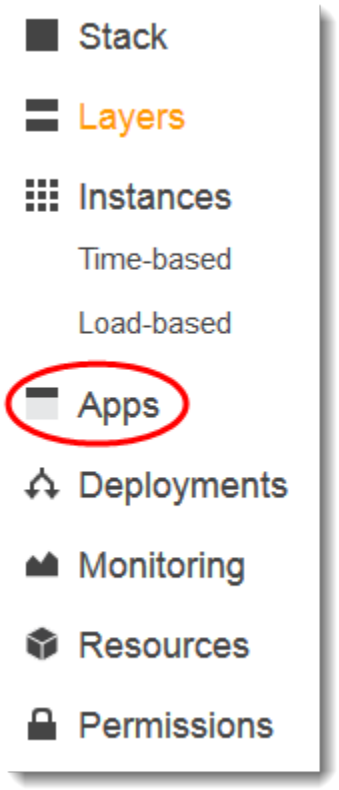
> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Informieren Sie AWS OpsWorks Stacks später in dieser exemplarischen Vorgehensweise über die App, die Sie auf der Instanz bereitstellen werden. In diesem Kontext definiert AWS OpsWorks Stacks eine App als Code, den Sie auf einer Instanz ausführen möchten. (Weitere Informationen finden Sie unter Apps.)

Das Verfahren in diesem Abschnitt gilt für Chef 12 und neuere Stacks. Weitere Informationen darüber, wie Anwendungen Ebenen in Chef 11-Stacks hinzugefügt werden, finden Sie unter Schritt 2.4: Erstellen und Bereitstellen einer Anwendung – Chef 11.

So legen Sie die Anwendung für die Bereitstellung fest

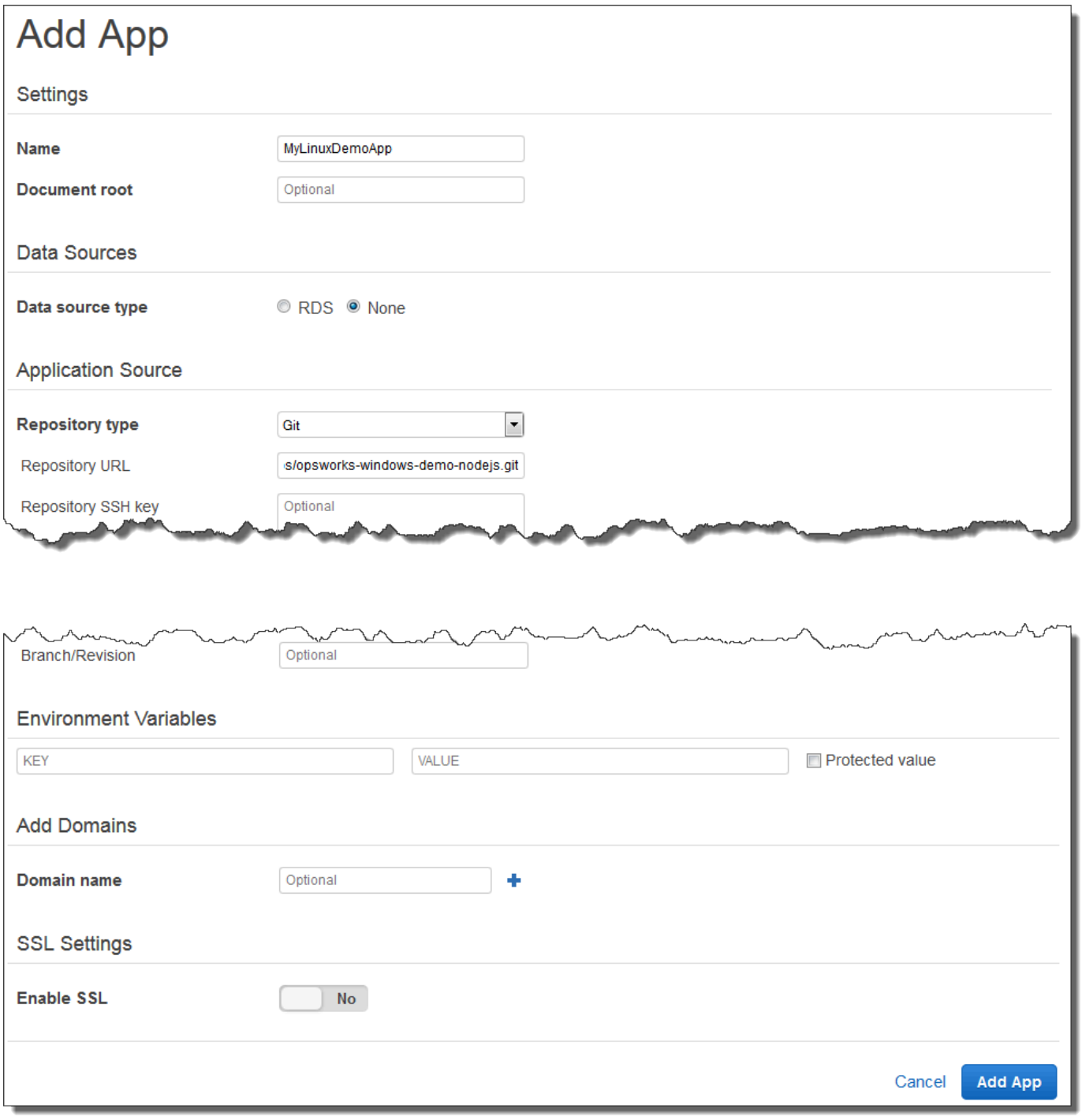
1. Wählen Sie im Service-Navigationsbereich Apps (Anwendungen) aus:



2. Die Seite Apps (Anwendungen) wird angezeigt. Wählen Sie Add an app (App hinzufügen) aus. Die Seite Add App (Anwendung hinzufügen) wird angezeigt.
3. Geben Sie unter Settings (Einstellungen) für Name den Namen **`MyLinuxDemoApp`** ein. (Sie können auch einen anderen Namen eingeben. Stellen Sie jedoch sicher, diesen anstelle von `MyLinuxDemoApp` zu nutzen.)
4. Geben Sie bei Application Source (Anwendungsquelle), für Repository URL (Repository-URL) die URL **`https://github.com/awslabs/opsworks-windows-demo-nodejs.git`** ein.
5. Übernehmen Sie die Standardeinstellungen für die folgenden Schritte:

   - Settings (Einstellungen), Document root (Basisverzeichnis) (leer)
   - Data Sources (Datenquelle), Data source type (Datenquellentyp) (None (Kein))
   - Repository type (Repository-Typ) (Git)
   - Repository SSH key (Repository-SSH-Schlüssel) (leer)
   - Branch/Revision (Zweig/Version) (leer)
   - Environment Variables (Umgebungsvariablen) (leer KEY (SCHLÜSSEL), leer VALUE (WERT), deaktiviert Protected Value (Geschützer Wert))
   - Add Domains (Domänen hinzufügen), Domain Name (Domänenname) (leer)

- SSL Settings (SSL-Einstellungen), Enable SSL (SSL aktivieren) (No (Nein))



6. Wählen Sie App hinzufügen. AWS OpsWorks Stacks fügt die App hinzu und zeigt die Apps-Seite an.

Sie haben nun eine Anwendung mit den richtigen Einstellungen für diese Anleitung.

Im nächsten Schritt starten Sie die Instance.

Schritt 5: Starten einer Instance

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

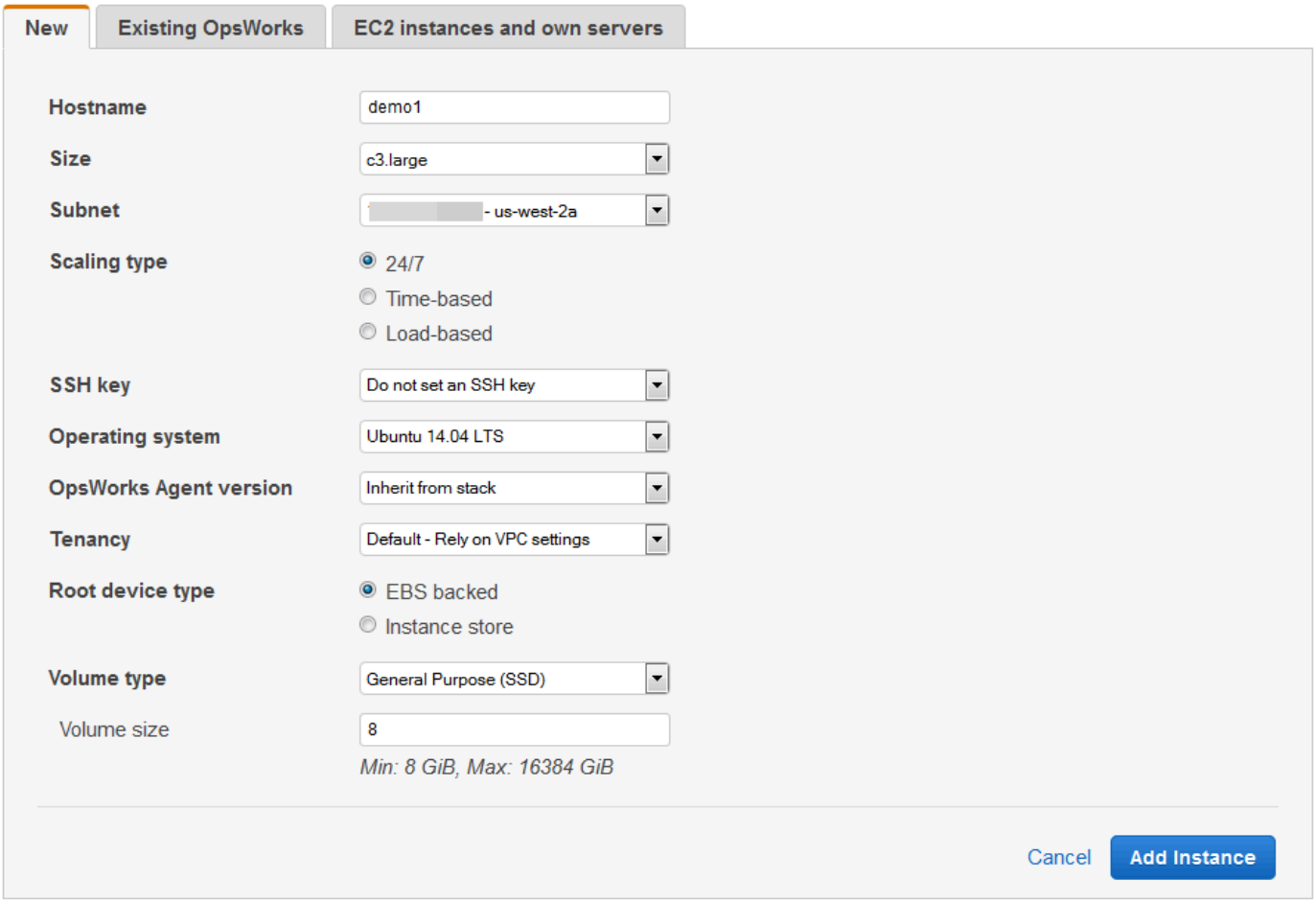
Verwenden Sie AWS OpsWorks Stacks, um eine Ubuntu Server Amazon EC2 EC2-Instance zu starten. Diese Instance verwendet die Einstellungen, die Sie in dem Layer festgelegt haben, den Sie zuvor in dieser Anleitung erstellt haben. (Weitere Informationen finden Sie unter Instances.)

So starten Sie die Instance

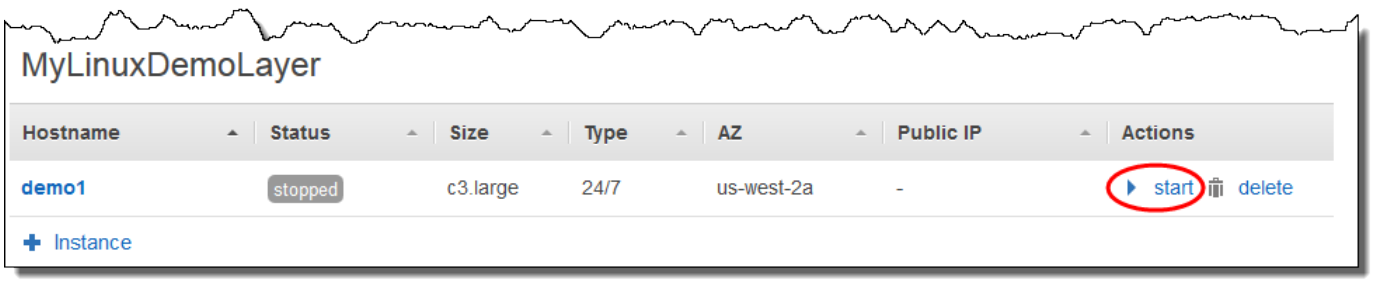
1. Wählen Sie im Service-Navigationsbereich Instances aus. Die Seite Instances wird angezeigt.
2. Wählen Sie für MyLinuxDemoLayerInstance hinzufügen aus.
3. Behalten Sie auf der Registerkarte New (Neu) die folgenden Standardeinstellungen bei:
   - Hostname (Hostname) (demo1 (demo1))
   - Size (c3.large)
   - Subnet (*IP-Adresse* us-west-2a)
4. Wählen Sie Advanced (Erweitert) aus.
5. Übernehmen Sie die Standardeinstellungen für die folgenden Schritte:
   - Scaling type (24/7)
   - SSH key (Do not use a default SSH key)
   - Operating system (Ubuntu 18.04 LTS)
   - OpsWorks Agentenversion (Vom Stapel erben)
   - Tenancy (Default - Rely on VPC settings)
   - Root device type (EBS backed)

- Volume type (General Purpose (SSD))
- Volume size (8)

6. Ihre Ergebnisse sollten ähnlich wie im folgenden Screenshot aussehen:



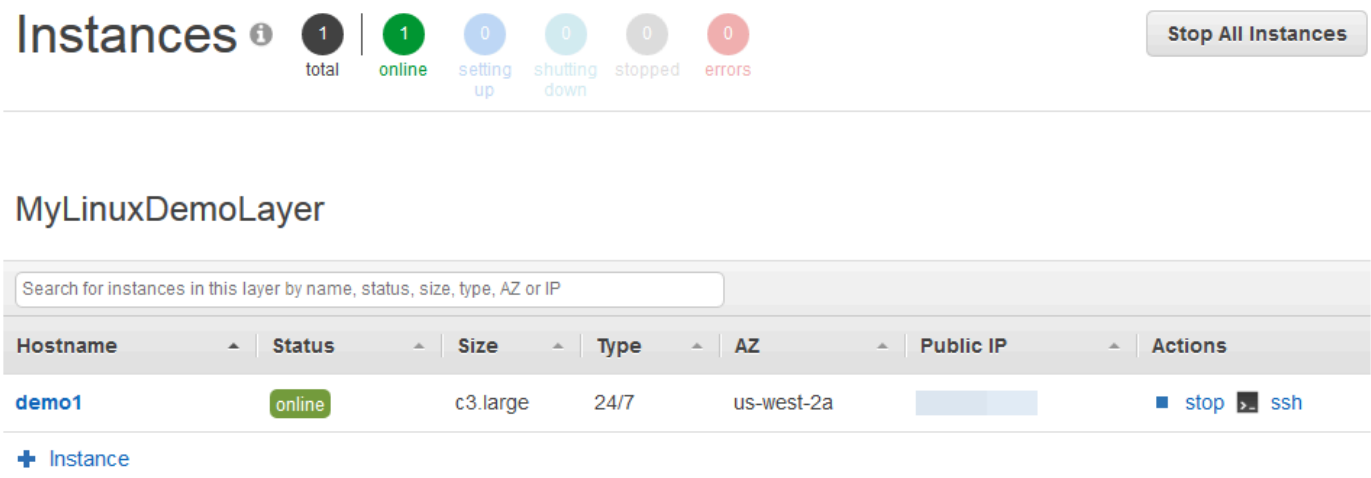
7. Wählen Sie Instanz hinzufügen. AWS OpsWorks Stacks fügt die Instanz dem Layer hinzu und zeigt die Seite „Instanzen“ an.
8. Wählen Sie für MyLinuxDemoLayer, für demo1, für Actions, Start:



9. Nach mehreren Minuten geschieht Folgendes:

   - Der Kreis setting up (Wird eingerichtet) ändert sich von 0 auf 1.
   - Status wechselt von stopped (Angehalten) zu requested (Angefragt) zu pending (Ausstehend) zu booting (Wird gebootet) zu running_setup (Einrichtung wird ausgeführt) und schließlich zu online. Beachten Sie, dass dieser Vorgang einige Minuten dauern kann.
   - Nachdem sich der Status in online geändert hat, ändert sich die Kreisanzeige setting up (Wird eingerichtet) von 1 in 0 und der online-Kreis von 0 in 1 und wechselt zu hellgrün. Fahren Sie nicht fort, bevor der online (Online)-Kreis hellgrün angezeigt wird und 1 Instance online anzeigt.

10. Bevor Sie fortfahren, muss Ihr Ergebnis wie auf der Abbildung dargestellt aussehen. Falls eine Fehlermeldung angezeigt wird, lesen Sie unter Handbuch zur Fehlersuche und Fehlerbehebung weiter:



Jetzt haben Sie eine Instance, die bereit ist, für die Anwendung bereitgestellt zu werden.

> **Note**
>
> Wenn Sie sich bei der Instance anmelden möchten, um sie weiter zu erkunden, müssen Sie AWS OpsWorks Stacks zunächst Informationen über Ihren öffentlichen SSH-Schlüssel zur Verfügung stellen (den Sie mit Tools wie ssh-keygen oder PuTTYgen erstellen können) und dann müssen Sie Berechtigungen für den `MyLinuxDemoStack` Stack festlegen, damit sich Ihr Benutzer bei der Instance anmelden kann. Anweisungen finden Sie unter Registrierung des öffentlichen SSH-Schlüssels eines Benutzers und Anmelden mit SSH. Wenn Sie SSH verwenden möchten, um über PuTTY eine Verbindung zu Instanzen herzustellen, finden

Sie in der Dokumentation Informationen unter Herstellen einer Verbindung zu Ihrer Linux-Instance von Windows aus. AWS

Im nächsten Schritt stellen Sie die Anwendung für die Instance bereit.

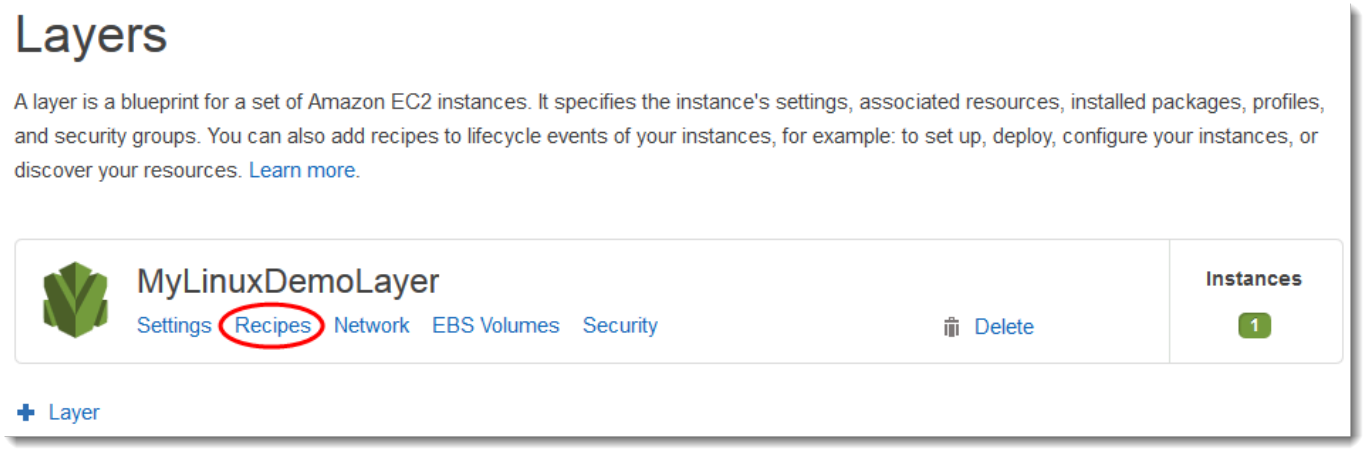
Schritt 6: Bereitstellen der Anwendung für die Instance

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Schritt werden Sie die App von der laufenden Instanz aus GitHub bereitstellen. (Weitere Informationen finden Sie unter Bereitstellen von Anwendungen.) Bevor Sie die Anwendung bereitstellen, müssen Sie das zu verwendende Rezept zur Koordinierung der Bereitstellung auswählen. Ein Rezept ist ein Chef-Konzept. Rezepte sind Anweisungen, geschrieben in Ruby-Sprachsyntax, die die Ressourcen für die Nutzung auswählen und die Reihenfolge bestimmen, in der diese Ressourcen angewendet werden. (Weitere Informationen finden Sie unter About Recipes auf der Website Learn Chef.)

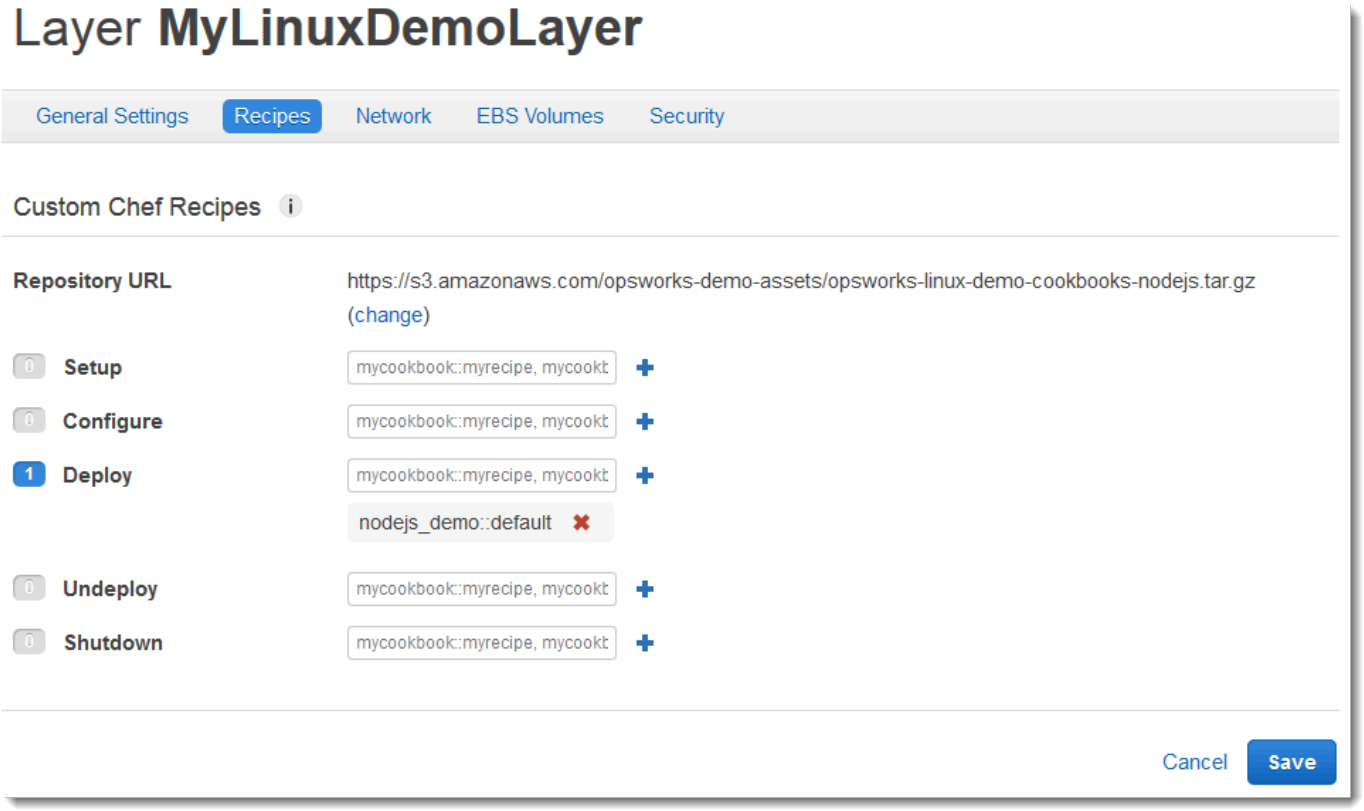
So legen Sie das Rezept für die Bereitstellung der Anwendung für die Instance fest

1. Wählen Sie im Service-Navigationsbereich Layers aus. Die Seite Layers wird angezeigt.
2. Wählen MyLinuxDemoLayerSie für Rezepte:



Die MyLinuxDemoLayerLayer-Seite wird mit geöffnetem Tab „Rezepte“ angezeigt.

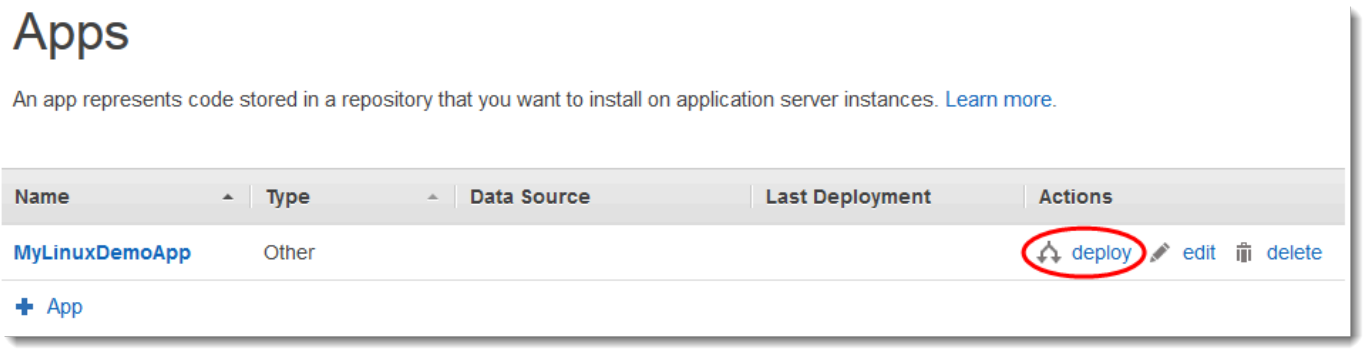
3. Geben Sie bei Custom Chef Recipes (Benutzerdefinierte Chef-Rezepte) für Deploy (Bereitstellen) die Zeichenfolge **`nodejs_demo::default`** ein und drücken Sie dann die Eingabetaste. `nodejs_demo` ist der Name des Rezeptbuches und `default` ist der Name des Zielrezepts innerhalb des Rezeptbuches. (Wenn Sie sich einen Überblick über die Rezept-Codes verschaffen möchten, lesen Sie Weiterführende Informationen: Arbeiten mit dem Rezeptbuch, das in dieser Anleitung verwendet wird.) Ihre Ergebnisse müssen wie auf dem folgenden Screenshot abgebildet aussehen:



4. Wählen Sie „Speichern“. AWS OpsWorks Stacks fügt das Rezept zum Deploy-Lifecycle-Ereignis des Layers hinzu.

So stellen Sie die Anwendung für die Instance bereit

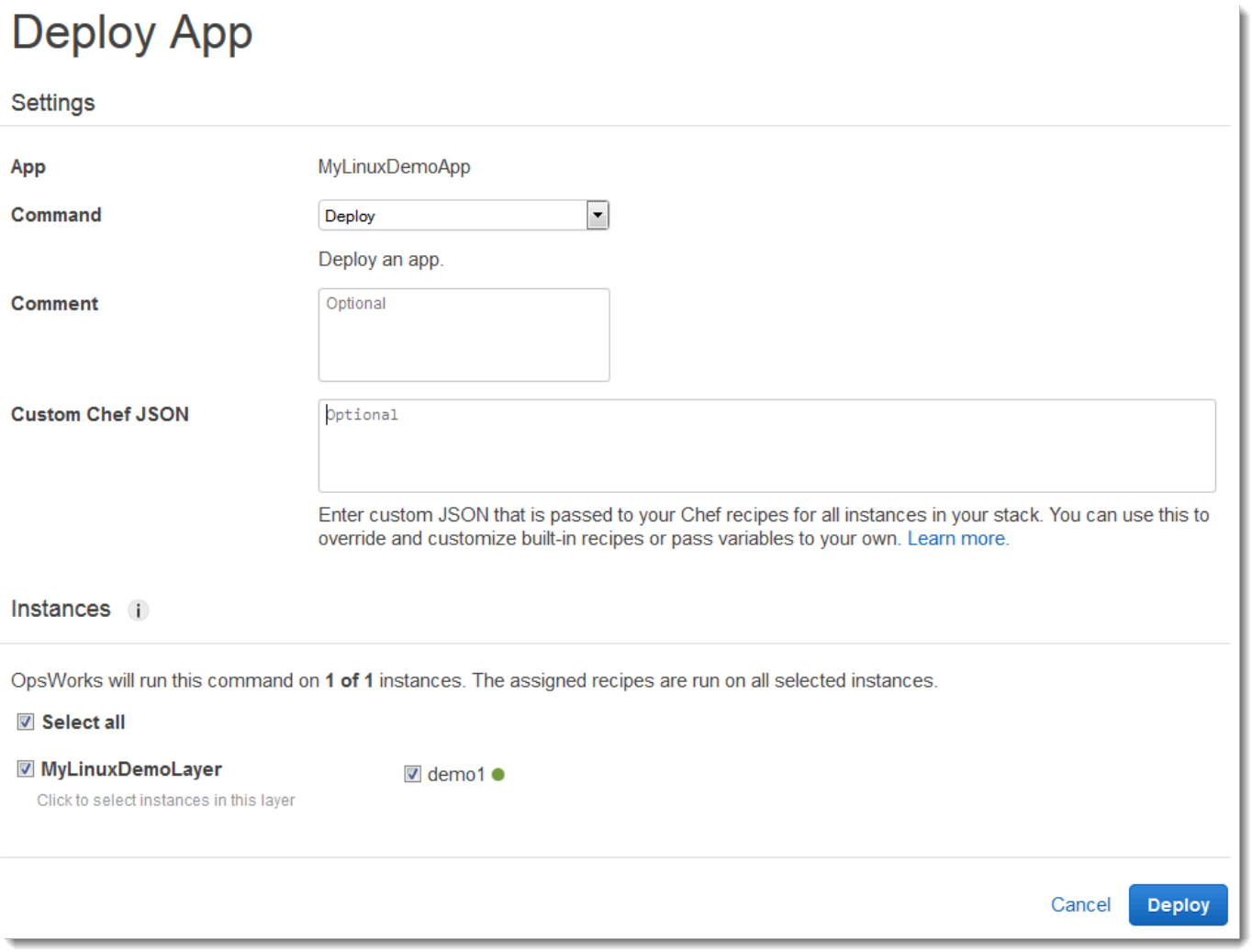
1. Wählen Sie im Service-Navigationsbereich Apps (Anwendungen) aus. Die Seite Apps (Anwendungen) wird angezeigt.
2. Wählen Sie für Aktionen die Option Deploy aus, wie im folgenden Screenshot dargestellt: MyLinuxDemoApp



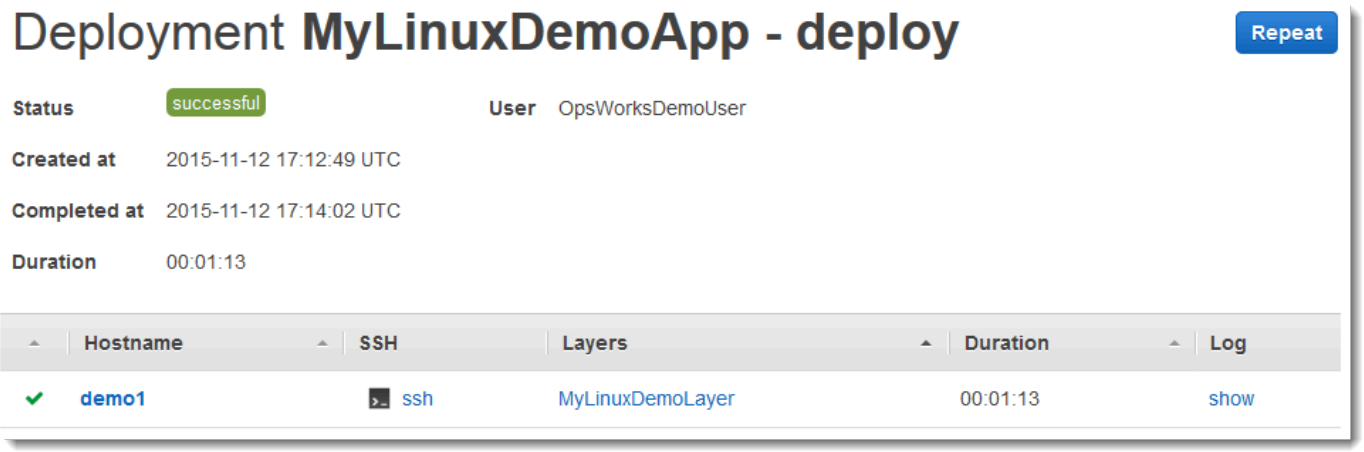
3. Behalten Sie auf der Seite Deploy App (Anwendung bereitstellen) die folgenden Standardeinstellungen bei:

   - Command (Befehl) (Deploy (Bereitstellen))
   - Comment (Kommentar) (leer)
   - Settings (Einstellungen), Advanced (Erweitert), Custom Chef JSON (Benutzerdefinierte JSON-Chef-Datei) (leer)
   - Instanzen, Erweitert (aktiviert Alle auswählen, aktiviert MyLinuxDemoLayer, als demo1 markiert)

4. Ihre Ergebnisse müssen wie auf dem folgenden Screenshot abgebildet aussehen:



5. Wählen Sie Bereitstellen. Die Seite Deployment MyLinuxDemoApp — Deploy wird angezeigt. Status ändert sich von running (Wird ausgeführt) in successful (Erfolgreich). Ein rotierender Kreis wird neben demo1 (demo1) angezeigt, der dann zu einem grünen Häkchen wird. Beachten Sie, dass dieser Vorgang einige Minuten dauern kann. Fahren Sie nicht fort, bis Status (Status) den Wert successful (Erfolgreich) hat und das grüne Häkchen-Symbol zu sehen ist.
6. Die Ergebnisse müssen mit dem folgenden Screenshot übereinstimmen, außer natürlich für Created at (Erstellt um), Completed at (Abgeschlossen um), Duration (Dauer) und User (Benutzer). Wenn status (Status) auf failed (Fehler) gesetzt ist, wählen Sie zur Fehlerbehebung für Log (Protokoll) die Option show (Anzeigen) aus, um Fehlerdetails zu erhalten:



Sie haben die Anwendung nun erfolgreich auf der Instance bereitgestellt.

Im nächsten Schritt werden Sie die bereitgestellte Anwendung auf der Instance testen.

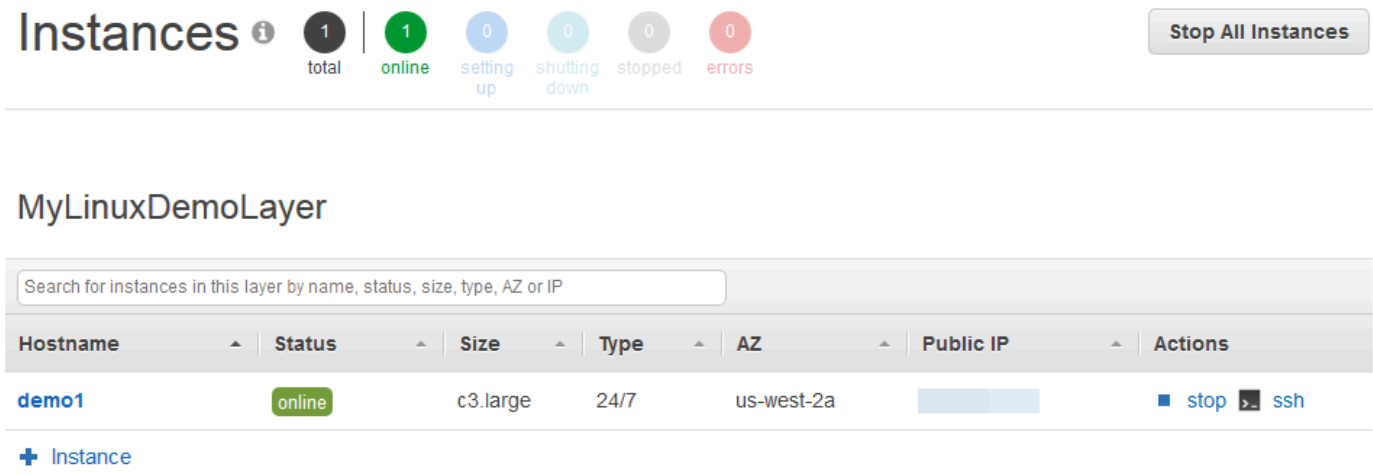
Schritt 7: Testen der bereitgestellten Anwendung auf der Instance

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

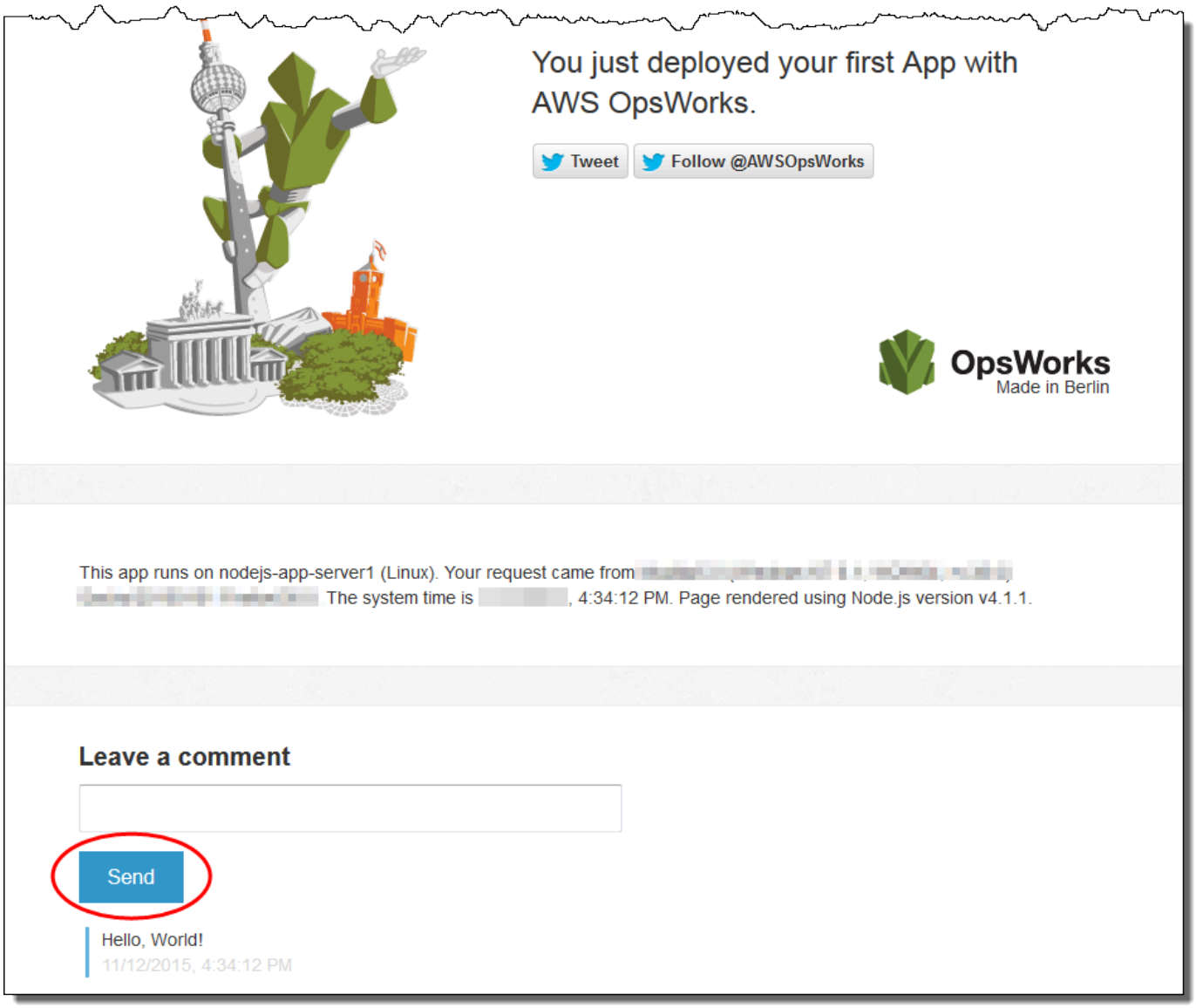
Jetzt testen Sie die Anwendungsbereitstellung auf der Instance.

So testen Sie die Bereitstellung auf der Instance

1. Wählen Sie im Service-Navigationsbereich Instances aus. Die Seite Instances wird angezeigt.
2. Wählen Sie für MyLinuxDemoLayer, für demo1, für Public IP, die IP-Adresse aus:



   Eine neue Registerkarte des Webbrowsers zeigt die Anwendung an.
3. Geben Sie auf der Glückwunsch-Webseite im Textfeld Leave a comment (Kommentar eingeben) einen Kommentar ein und wählen Sie Send (Senden) aus, um die Anwendung zu testen. Die Anwendung fügt den Kommentar zur Webseite hinzu. Sie können beliebig oft Kommentare hinterlassen und Send (Senden) auswählen:



4. Wenn du ein Twitter-Konto hast, wähle Tweet oder Follow @ und folge den Anweisungen auf dem BildschirmAWSOpsWorks, um über die App zu twittern oder @ zu folgen. AWSOpsWorks

Sie haben jetzt die bereitgestellte Anwendung erfolgreich auf der Instance getestet.

Im nächsten Schritt können Sie die AWS Ressourcen bereinigen, die Sie für diese exemplarische Vorgehensweise verwendet haben. Dieser nächste Schritt ist optional. Möglicherweise möchten Sie diese AWS Ressourcen weiterhin verwenden, wenn Sie mehr über AWS OpsWorks Stacks erfahren. Wenn Sie diese AWS Ressourcen behalten, kann dies jedoch zu laufenden Gebühren für Ihr AWS Konto führen. Wenn Sie diese AWS Ressourcen für eine spätere Verwendung behalten möchten,

haben Sie diese exemplarische Vorgehensweise nun abgeschlossen, und Sie können Nächste Schritte weitermachen.

Schritt 8 (Optional): Bereinigen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Um zu verhindern, dass zusätzliche Gebühren für Ihr AWS Konto anfallen, können Sie die AWS Ressourcen löschen, die für diese Komplettlösung verwendet wurden. Zu diesen AWS Ressourcen gehören der AWS OpsWorks Stacks-Stack und die Komponenten des Stacks. (Weitere Informationen finden Sie unter AWS OpsWorks Preisgestaltung.) Möglicherweise möchten Sie diese AWS Ressourcen jedoch weiterhin nutzen, um mehr über AWS OpsWorks Stacks zu erfahren. Wenn Sie diese AWS Ressourcen weiterhin verfügbar halten möchten, haben Sie diese exemplarische Vorgehensweise nun abgeschlossen, und Sie können weitermachen. Nächste Schritte

Inhalte, die in den Ressourcen gespeichert sind, die Sie für diese schrittweise Anleitung erstellt haben, können persönlich identifizierende Informationen enthalten. Wenn Sie nicht mehr möchten, dass diese Informationen von AWS gespeichert werden, führen Sie die in diesem Thema beschriebenen Schritte aus.

So löschen Sie die Anwendung aus dem Stack

1. Wählen Sie in der AWS OpsWorks Stacks-Konsole im Servicenavigationsbereich Apps aus. Die Seite Apps (Anwendungen) wird angezeigt.
2. Wählen Sie für Aktionen die Option Löschen aus. MyLinuxDemoApp Wenn die Bestätigungsmeldung angezeigt wird, wählen Sie Löschen. AWS OpsWorks Stacks löscht die App.

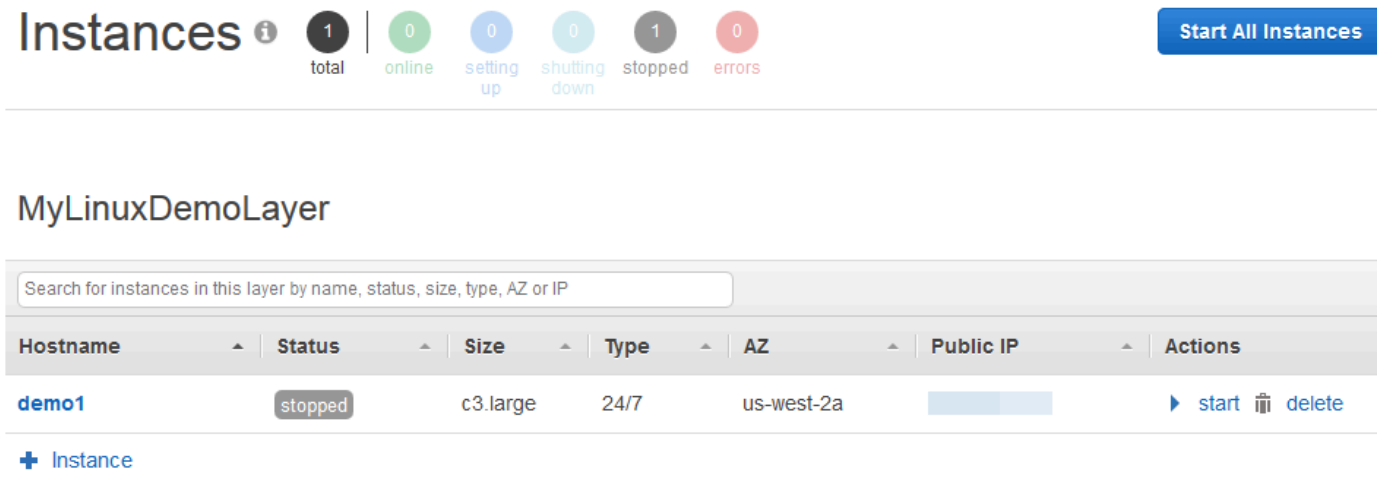
So löschen Sie die Instance für den Stack

1. Wählen Sie im Service-Navigationsbereich Instances aus. Die Seite Instances wird angezeigt.

2. Wählen Sie für MyLinuxDemoLayer, für demo1, für Actions, Stopp. Wählen Sie im Bestätigungsdialogfeld Stop aus. Es geschieht Folgendes.

   - Der Status ändert sich von online zu stopping (Wird angehalten) und schließlich zu stopped (Angehalten).
   - online ändert sich von 1 zu 0.
   - shutting down (Wird heruntergefahren) ändert sich von 0 zu 1 und schließlich wieder zu 0.
   - stopped ändert sich schließlich von 0 zu 1.

   Dieser Vorgang kann einige Minuten dauern. Wenn AWS OpsWorks Stacks fertig ist, werden die folgenden Ergebnisse angezeigt.



3. Wählen Sie bei Actions (Aktionen) die Option delete (löschen) aus. Wenn Sie die Bestätigungsmeldung sehen, wählen Sie Löschen. AWS OpsWorks Stacks löscht die Instanz und zeigt die Meldung Keine Instanzen an.

So löschen Sie den Stack

1. Wählen Sie im Service-Navigationsbereich Stack aus. Die Seite MyLinuxDemoStack wird angezeigt.
2. Wählen Sie Delete Stack. Wenn Sie die Bestätigungsmeldung sehen, wählen Sie Löschen. AWS OpsWorks Stacks löscht den Stapel und zeigt die OpsWorks Dashboard-Seite an.

Optional können Sie das Benutzer- und Amazon EC2 EC2-Schlüsselpaar, das Sie für diese exemplarische Vorgehensweise verwendet haben, löschen, wenn Sie sie nicht für den Zugriff auf

andere AWS Services und EC2-Instances wiederverwenden möchten. Anweisungen finden Sie unter Löschen eines IAM-Benutzers und Amazon EC2 EC2-Schlüsselpaare und Linux-Instances.

Sie haben diese Anleitung nun abgeschlossen. Weitere Informationen finden Sie unter Nächste Schritte.

Nächste Schritte

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Nachdem Sie diese exemplarische Vorgehensweise abgeschlossen haben, können Sie mehr über die Verwendung von Stacks erfahren: AWS OpsWorks

- Erfahren Sie mehr über das Rezeptbuch und die Anwendung, die Sie in dieser Anleitung verwendet haben. Siehe Weiterführende Informationen: Arbeiten mit dem Rezeptbuch, das in dieser Anleitung verwendet wird und Weiterführende Informationen: Arbeiten mit der Anwendung, die in dieser Anleitung verwendet wird.
- Üben Sie die Verwendung von AWS OpsWorks Stacks mit Windows-Instanzen. Siehe Erste Schritte: Windows.
- Weitere Informationen zu Stacks finden Sie auch unter Erstellen eines neuen Stacks.
- Weitere Informationen zu Layern finden Sie unter Bearbeiten der Konfiguration einer Ebene OpsWorks .
- Weitere Informationen zu Instances finden Sie unter Hinzufügen einer Instance zu einem Layer.
- Weitere Informationen zu Apps finden Sie unter Bereitstellen von Anwendungen.
- Weitere Informationen zu Cookbooks und Rezepte.
- Erstellen Sie Ihre eigenen Rezeptbücher. Siehe Erste Schritte: Rezeptbücher.
- Weitere Informationen zur Zugriffssteuerung für Stacks finden Sie unter Sicherheit und Berechtigungen.

Weiterführende Informationen: Arbeiten mit dem Rezeptbuch, das in dieser Anleitung verwendet wird

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Thema wird das Kochbuch beschrieben, das AWS OpsWorks Stacks für die Komplettlösung verwendet hat.

Ein Rezeptbuch ist ein Chef-Konzept. Rezeptbücher sind Archivdateien mit Konfigurationsinformationen, z. B. Rezepten, Attributwerten, Dateien, Vorlagen, Bibliotheken, Definitionen und benutzerdefinierten Ressourcen. Ein Rezept ist auch ein Chef-Konzept. Rezepte sind Anweisungen, geschrieben in Ruby-Sprachsyntax, die die Ressourcen für die Nutzung auswählen und die Reihenfolge bestimmen, in der diese Ressourcen angewendet werden. Weitere Informationen finden Sie unter About Cookbooks und About Recipes auf der Website Learn Chef.

Um den Inhalt des in dieser exemplarischen Vorgehensweise verwendeten Kochbuches zu sehen, extrahieren Sie den Inhalt der Datei opsworks-linux-demo-cookbooks-nodejs.tar.gz in ein leeres Verzeichnis auf Ihrer lokalen Workstation. (Sie können sich auch in der Instance anmelden, auf der Sie das Rezeptbuch bereitgestellt haben, und sich mit den Inhalten des Verzeichnisses `/var/chef/cookbooks` vertraut machen.)

Das Rezeptbuch führt den Code in der Datei `default.rb` im Verzeichnis `cookbooks/nodejs_demo/recipes` aus:

```
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
```

Die Datei geht folgendermaßen vor:

- `search(:aws_opsworks_app).first` verwendet die Chef-Suche, um Informationen über die Anweisung zu suchen, die letztendlich für die Instance bereitgestellt wird. Diese Information umfasst Einstellungen wie die Kurzbezeichnung der Anwendung und die Details ihres Quell-Repositorys. Da nur eine Anwendung in dieser Anleitung bereitgestellt wurde, bekommt die Chef-Suche diese Einstellungen von dem ersten Informationselement innerhalb des `aws_opsworks_app`-Suchindex auf der Instance. Immer wenn eine Instance gestartet wird, speichert AWS OpsWorks Stacks diese und andere zugehörige Informationen als Satz von Datenbeuteln auf der Instance selbst, und Sie erhalten den Inhalt der Datentasche über die Chef-Suche. Obwohl Sie diese Einstellungen in dieses Rezept fest programmieren können, ist die Nutzung von Data Bags und der Chef-Suche ein solideres Konzept. Weitere Informationen zu Data Bags finden Sie unter AWS OpsWorks Referenz für Stacks Data Bag. Sie können sich auch unter About Data Bags auf der Website Learn Chef informieren. Weitere Informationen über die Chef-Suche finden Sie unter About Search auf der Website Learn Chef.
- Die `package`-Ressource installiert Git auf der Instance.
- Die `application`-Ressource beschreibt und stellt Webanwendungen bereit:
  - `javascript`ist die Version der JavaScript Runtime, die installiert werden soll.
  - `environment` legt eine Umgebungsvariable fest.
  - `git` bekommt den Quellcode von dem angegebenen Repository und der Branch.
  - `app_path` ist der Pfad auf dem das Repository geklont werden soll. Wenn der Pfad auf der Instanz nicht existiert, erstellt AWS OpsWorks Stacks ihn.

- `link` erstellt einen symbolischen Link.
- `npm_install` installiert Node Package Manager, den standardmäßigen Paket-Manager für Node.js.
- `npm_start` führt Node.js aus.

AWS OpsWorks Stacks hat zwar das für diese Komplettlösung verwendete Kochbuch erstellt, Sie können jedoch auch Ihre eigenen Kochbücher erstellen. Um zu erfahren wie dies geht, vgl. Erste Schritte: Rezeptbücher. Weitere Informationen finden Sie unter About Cookbooks, About Recipes und Learn the Chef Basics on Ubuntu auf der Website Learn Chef und im Abschnitt "Our first Chef cookbook" unter First steps with Chef auf der Website Getting started with Chef.

Weiterführende Informationen: Arbeiten mit der Anwendung, die in dieser Anleitung verwendet wird

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Thema wird die App beschrieben, die AWS OpsWorks Stacks für diese exemplarische Vorgehensweise auf der Instanz bereitstellt.

Um den Quellcode der App zu sehen, extrahieren Sie den Inhalt des opsworks-windows-demo-nodejs GitHub Repositorys in ein leeres Verzeichnis auf Ihrer lokalen Workstation. Sie können sich auch in der Instance anmelden, auf der Sie das Rezeptbuch bereitgestellt haben, und sich mit den Inhalten des Verzeichnisses `/srv/mylinuxdemoapp` vertraut machen.

Die Datei `index.js` enthält den höchstwertigsten Code für die Anwendung:

```
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);
});
```

Die Datei geht folgendermaßen vor:

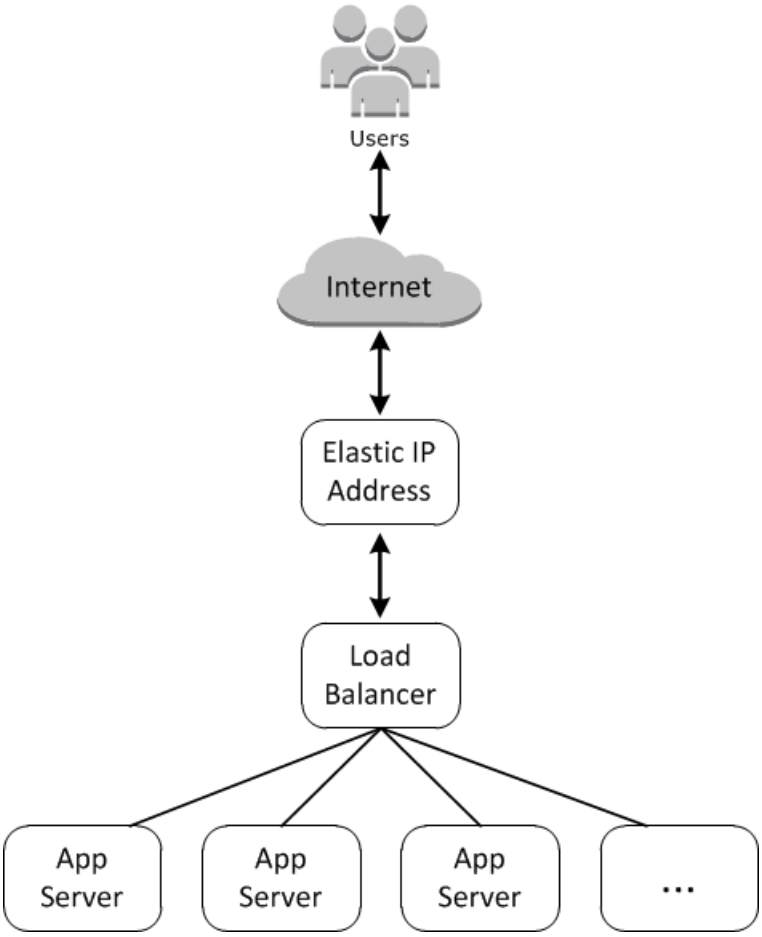
- `require` lädt Module mit einigen abhängigen Codes, die diese Webanwendung benötigt, um wie erwartet ausgeführt zu werden.
- Die Funktionen `add_comment` und `get_comments` schreiben Informationen in die Datei `comments.json` und entnehmen sie ihr auch.
- Weitere Informationen zu `app.get`, `app.listen`, `app.post`, `app.set` und `app.use` finden Sie unter Express API Reference.

Informationen dazu, wie Sie Ihre Anwendung für die Bereitstellung erstellen und packen, finden Sie unter Anwendungsquelle.

Erste Schritte mit Windows-Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Cloud-basierte Anwendungen erfordern in der Regel eine Gruppe verwandter Ressourcen — Anwendungsserver, Datenbankserver usw. —, die gemeinsam erstellt und verwaltet werden müssen. Diese Instances-Sammlung wird Stack genannt. Ein einfacher Anwendungs-Stack hat beispielsweise folgende Struktur.



Die grundlegende Architektur umfasst Folgendes:

- Einer Elastic IP-Adresse, um Benutzeranfragen zu empfangen
- Einem Load Balancer, um eingehende Anfragen gleichmäßig auf die Anwendungsserver zu verteilen
- So viele Anwendungsserver-Instances wie erforderlich, um den Datenverkehr handhaben zu können.

Darüber hinaus benötigen Sie in der Regel eine Methode, um Anwendungen auf den Anwendungsservern zu verteilen, Benutzerberechtigungen zu verwalten usw.

AWS OpsWorks Stacks bietet eine einfache und unkomplizierte Möglichkeit, Stacks und die zugehörigen Anwendungen und Ressourcen zu erstellen und zu verwalten. In diesem Kapitel werden die Grundlagen von AWS OpsWorks Stacks — zusammen mit einigen der komplexeren Funktionen — vorgestellt, indem Sie im Diagramm Schritt für Schritt durch den Prozess der Erstellung des Anwendungsserver-Stacks geführt werden. Es verwendet ein inkrementelles Entwicklungsmodell, das

mit AWS OpsWorks Stacks leicht nachzuvollziehen ist: Richten Sie einen Basisstapel ein und fügen Sie, nachdem er ordnungsgemäß funktioniert, Komponenten hinzu, bis Sie eine Implementierung mit vollem Funktionsumfang erhalten.

- Schritt 1: Erfüllen der Voraussetzungen erläutert die vorbereitenden Maßnahmen, um mit der Anleitung zu beginnen.
- Schritt 2: Erstellen eines Basis-Stacks für einen Anwendungsserverzeigt, wie Sie einen grundlegenden Stack zur Unterstützung von Internetinformationsdiensten (IIS) erstellen und eine App auf dem Server bereitstellen.
- Schritt 3: Skalieren von IISExample zeigt, wie Sie einen Stack skalieren, um zusätzliche Auslastung zu verarbeiten, indem Sie weitere Anwendungsserver, einen Load Balancer zur Verteilung des eingehenden Datenverkehrs und eine Elastic IP-Adresse zum Annehmen eingehender Anfragen hinzufügen.

Themen

Schritt 1: Erfüllen der Voraussetzungen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Um mit der Anleitung beginnen zu können, müssen Sie die folgenden Einrichtungsschritte ausführen. Zu diesen Einrichtungsschritten gehören die Registrierung für ein AWS Konto, die Erstellung eines Administratorbenutzers und die Zuweisung von Zugriffsberechtigungen für Stacks. AWS OpsWorks

Wenn Sie bereits eine der Anleitungen Erste Schritte: Beispiel oder Erste Schritte: Linux durchgearbeitet haben, erfüllen Sie bereits die Voraussetzungen für diese Anleitung und können direkt mit Schritt 2: Erstellen eines Basis-Stacks für einen Anwendungsserver fortfahren.

Themen

- Registriere dich für ein AWS-Konto
- Erstellen Sie einen Benutzer mit Administratorzugriff
- Weisen Sie Dienstzugriffsberechtigungen zu
- Stellen Sie sicher, dass AWS OpsWorks Stacks-Benutzer zu Ihrer Domain hinzugefügt wurden

Registriere dich für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu https://aws.amazon.com/ auf und klicken Sie auf Mein Konto.

Erstellen Sie einen Benutzer mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen Sie einen Benutzer mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis im AWS IAM Identity Center Benutzerhandbuch.

Melden Sie sich als Benutzer mit Administratorzugriff an

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal.

Weisen Sie weiteren Benutzern Zugriff zu

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

Anweisungen finden Sie im Benutzerhandbuch unter Einen Berechtigungssatz erstellen.AWS IAM Identity Center

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Anweisungen finden Sie im AWS IAM Identity Center Benutzerhandbuch unter Gruppen hinzufügen.

Weisen Sie Dienstzugriffsberechtigungen zu

Ermöglichen Sie den Zugriff auf den AWS OpsWorks Stacks-Dienst (und die zugehörigen Dienste, auf die AWS OpsWorks Stacks angewiesen ist), indem Sie Ihrer Rolle oder Ihrem `AmazonS3FullAccess` Benutzer die Berechtigungen `AWSOpsWorks_FullAccess` und hinzufügen.

Weitere Informationen zum Hinzufügen von Berechtigungen finden Sie unter Hinzufügen von IAM-Identitätsberechtigungen (Konsole).

Stellen Sie sicher, dass AWS OpsWorks Stacks-Benutzer zu Ihrer Domain hinzugefügt wurden

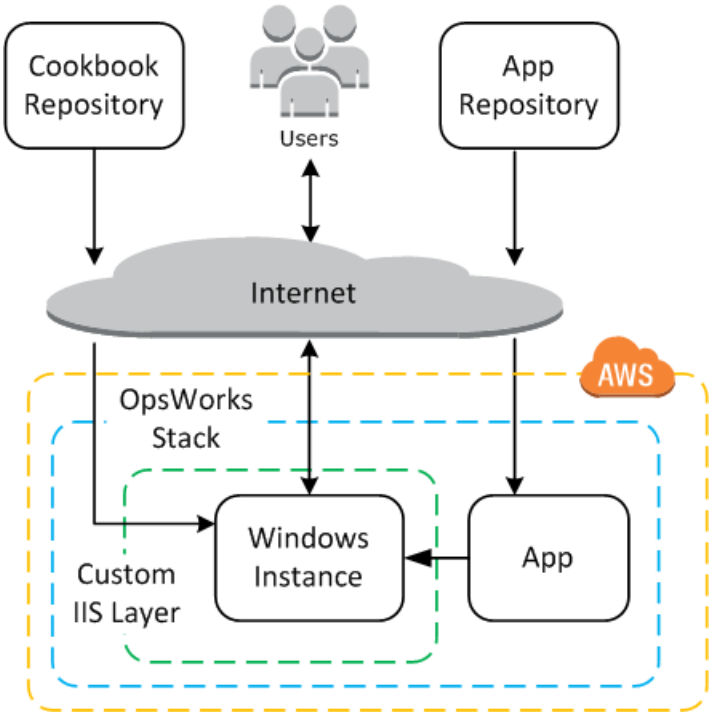
In einem Chef 12.2 Stack erstellt das enthaltene Rezeptbuch `aws_opsworks_users` Nutzer, die einen SSH- und Remote Desktop Protocol (RDP)-Zugriff auf Windows-basierte Instances haben. Wenn Sie Windows-Instanzen in Ihrem Stack mit einer Active Directory-Domäne verbinden, kann dieser Cookbook-Lauf fehlschlagen, wenn die AWS OpsWorks Stacks-Benutzer nicht in Active Directory existieren. Werden die Benutzer im Active Directory nicht erkannt, können Instances in einen `setup failed`-Status übergehen, wenn Sie sie nach dem Hinzufügen zu einer Domäne neu starten. Für mit einer Domäne verknüpfte Windows-Instances reicht es nicht aus, AWS OpsWorks Stacks-Benutzern den SSH-/RDP-Zugriff auf der Seite der Benutzerberechtigungen zu gewähren.

Bevor Sie Windows-Instanzen in einem Chef 12.2-Stack mit einer Active Directory-Domäne verbinden, stellen Sie sicher, dass alle AWS OpsWorks Stacks-Benutzer des Windows-basierten Stacks Mitglieder der Domäne sind. Der beste Weg, dies zu tun, besteht darin, die föderierte Identität mit IAM zu konfigurieren, bevor Sie Ihren Windows-basierten Stack erstellen, und dann Verbundbenutzer in AWS OpsWorks Stacks zu importieren, bevor Sie Instanzen in Ihrem Stack zu einer Domäne verbinden. Weitere Informationen dazu finden Sie unter Enabling Federation to AWS Using Windows Active Directory, ADFS, and SAML 2.0 im AWS-Sicherheitsblog und Federating Existing Users im IAM-Benutzerhandbuch.

Schritt 2: Erstellen eines Basis-Stacks für einen Anwendungsserver

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Ein Basis-Anwendungsserver-Stack besteht aus einer einzelnen Anwendungsserver-Instance mit einer öffentlichen IP-Adresse für den Empfang von Benutzeranforderungen. Der Anwendungscode und zugehörige Dateien werden in einem separaten Repository gespeichert und von dort auf dem Server bereitgestellt. Das folgende Diagramm veranschaulicht einen solchen Stack.



Der Stack besteht aus folgenden Komponenten:

- Einer Ebene, die eine Gruppe von Instances repräsentiert und festlegt, wie diese konfiguriert werden.

  Der Layer in diesem Beispiel steht für eine Gruppe von IIS-Instances.
- Eine Instance, die eine Amazon EC2 EC2-Instance darstellt.

In diesem Fall konfiguriert der Layer eine einzelne Instance, um IIS auszuführen. Layer können jedoch auch mehrere Instances haben.

- Eine Anwendung, die die erforderlichen Informationen zur Installation einer Anwendung in der Instance enthält.
- Ein Rezeptbuch mit benutzerdefinierten Chef-Rezepten, die benutzerdefinierte IIS-Ebene unterstützen. Das Kochbuch und der App-Code werden in Remote-Repositorys gespeichert, z. B. in einer Archivdatei in einem Amazon S3 S3-Bucket oder einem Git-Repository.

In den folgenden Abschnitten wird beschrieben, wie Sie die AWS OpsWorks Stacks-Konsole verwenden, um den Stack zu erstellen und die Anwendung bereitzustellen.

Themen

Schritt 2.1: Erstellen des Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie starten ein AWS OpsWorks Stacks-Projekt, indem Sie einen Stack erstellen, der als Container für Ihre Instances und andere Ressourcen fungiert. Die Stack-Konfiguration legt einige grundlegende Einstellungen fest, wie z. B. die AWS-Region und das Standardbetriebssystem, die von allen Stack-Instances gemeinsam genutzt werden.

Erstellen eines neuen Stacks

1. Hinzufügen eines Stacks

   Falls Sie sich noch nicht bei der AWS OpsWorks Stacks-Konsole angemeldet haben, tun Sie das jetzt.

   - Wenn für das Konto keine vorhandenen Stacks vorhanden sind, wird die OpsWorks Seite Willkommen bei AWS angezeigt. Wählen Sie Add your first stack aus.
   - Andernfalls wird das AWS OpsWorks Stacks-Dashboard angezeigt, in dem die Stacks Ihres Kontos aufgeführt sind. Wählen Sie Stack hinzufügen.

2. Konfigurieren des Stacks

   Wählen Sie auf der Seite Add Stack (Stack hinzufügen) die Option Chef 12 stack (Chef 12-Stack) aus und geben Sie die folgenden Einstellungen an:

   Stack name

   Geben Sie einen Namen für Ihren Stack ein, der alphanumerische Zeichen (a—z, A—Z und 0—9) und Bindestriche (-) enthalten kann. Der Beispiel-Stack in dieser Anleitung hat den Namen **`IISWalkthrough`**.

   Region

   Wählen Sie US West (Oregon) als Region des Stacks aus.

   Sie können einen Stack in jeder Region erstellen, wir empfehlen jedoch US West (Oregon) für Tutorials.

   Standard-Betriebssystem

   Wählen Sie Windows aus, und geben Sie dann Microsoft Windows Server 2022 Base an, was die Standardeinstellung ist.

   Verwenden von benutzerdefinierten Chef-Rezeptbüchern

   Wählen Sie für diese Anleitung den Wert No (Nein) aus.

3. Wählen Sie Advanced (Erweitert) aus, um zu bestätigen, dass Sie über die IAM-Rolle verfügen und das Standard-IAM-Instance-Profil ausgewählt ist.

IAM-Rolle

Geben Sie die IAM (AWS Identity and Access Management) -Rolle des Stacks an. AWS OpsWorks Stacks muss auf andere AWS-Services zugreifen, um Aufgaben wie das Erstellen und Verwalten von Amazon EC2 EC2-Instances auszuführen. Die IAM-Rolle gibt die Rolle an, die AWS OpsWorks Stacks annimmt, um in Ihrem Namen mit anderen AWS-Services zusammenzuarbeiten. Weitere Informationen finden Sie unter AWS OpsWorks Stacks erlauben, in Ihrem Namen zu handeln.

- Wenn Ihr Konto bereits über eine AWS OpsWorks Stacks-IAM-Rolle verfügt, können Sie diese aus der Liste auswählen.

  Wenn die Rolle von AWS OpsWorks Stacks erstellt wurde, erhält sie einen Namen. `aws-opsworks-service-role`

- Wählen Sie andernfalls Neue IAM-Rolle aus, um AWS OpsWorks Stacks anzuweisen, eine neue Rolle mit den richtigen Berechtigungen für Sie zu erstellen.

  Hinweis: Wenn Sie umfassende Zugriffsberechtigungen für AWS OpsWorks Stacks haben, benötigen Sie für das Erstellen einer neuen Rolle einige zusätzliche IAM-Berechtigungen. Weitere Informationen finden Sie unter Beispielrichtlinien.

4. Übernehmen Sie die Standardwerte für die restlichen Einstellungen und wählen Sie Add Stack (Stack hinzufügen) aus. Weitere Informationen zu den verschiedenen Stack-Einstellungen finden Sie unter Erstellen eines neuen Stacks.

Schritt 2.2: Autorisieren von RDP-Zugriff

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Nachdem Sie einen Stack erstellt haben, erstellen Sie nun einen Layer und fügen diesem eine Windows-Instance hinzu. Zunächst müssen Sie jedoch den Stack so konfigurieren, dass RDP für die

Verbindung mit den benutzerdefinierten Instances des Layers verwendet wird. Gehen Sie dazu wie folgt vor:

- Fügen Sie der Sicherheitsgruppe, die für den RDP-Zugriff zuständig ist, eine Regel für eingehenden Datenverkehr hinzu.
- Lege deine AWS OpsWorks Stacks-Berechtigungen für diesen Stack fest, um RDP-Zugriff zu ermöglichen.

Wenn Sie den ersten Stack in einer Region erstellen, erstellt AWS OpsWorks Stacks eine Reihe von Sicherheitsgruppen. Dazu gehört eine mit dem Namen etwa`AWS-OpsWorks-RDP-Server`, die AWS OpsWorks Stacks an alle Windows-Instanzen anhängt, um RDP-Zugriff zu ermöglichen. Standardmäßig sind in diesen Sicherheitsgruppe jedoch keine Regeln enthalten. Daher müssen Sie eine Regel für den eingehenden Datenverkehr zum Zulassen von RDP-Zugriff auf Ihre Instances hinzufügen.

So ermöglichen Sie den RDP-Zugriff

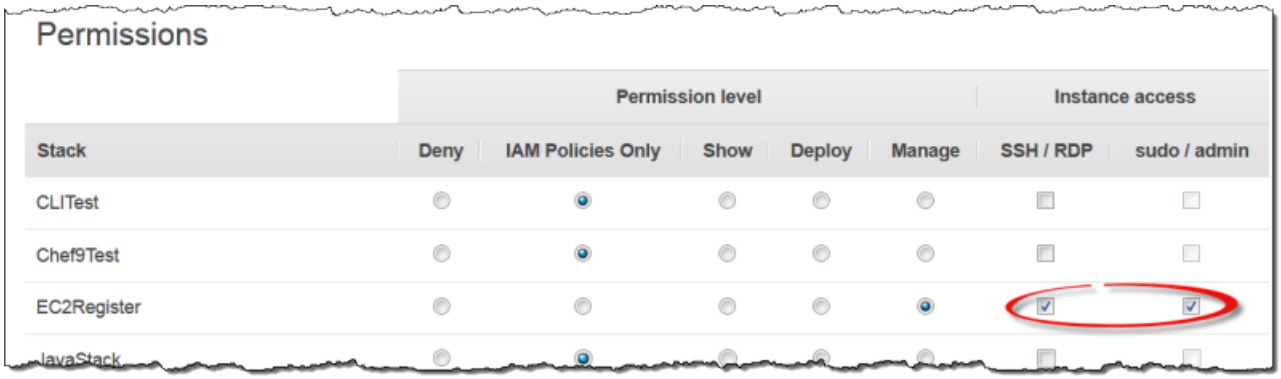
1. Öffnen Sie die Amazon EC2 EC2-Konsole, stellen Sie sie auf die Region des Stacks ein und wählen Sie im Navigationsbereich Sicherheitsgruppen aus.
2. Wählen Sie AWS- OpsWorks -RDP-Server, klicken Sie auf die Registerkarte Inbound und dann auf Bearbeiten.
3. Wählen Sie Add Rule (Regel hinzufügen) aus und legen Sie die folgenden Einstellungen fest:
   - Typ — RDP.
   - Quelle — Die zulässigen Quell-IP-Adressen.

     In der Regel erlauben Sie eingehende RDP-Anfragen von Ihrer eigenen IP-Adresse oder einem festen IP-Adressbereich (üblicherweise der IP-Adressbereich Ihres Unternehmens). Für diese Anleitung können Sie einfach 0.0.0.0/0 verwenden, um RDP-Zugriff von beliebigen IP-Adressen zuzulassen.

Die Sicherheitsgruppe ermöglicht der Instance, RDP-Verbindungsanfragen zu empfangen. Das ist jedoch noch nicht alles. Ein normaler Benutzer meldet sich mit einem von AWS OpsWorks Stacks bereitgestellten Passwort bei der Instanz an. Damit AWS OpsWorks Stacks dieses Passwort generiert, müssen Sie den RDP-Zugriff für den Benutzer explizit autorisieren.

So autorisieren Sie RDP-Zugriff für einen Benutzer

1. Wählen Sie im AWS OpsWorks Stacks-Dashboard den IISWalkthrough-Stack aus.
2. Wählen Sie im Navigationsbereich des Stacks Permissions (Berechtigungen) aus.
3. Wählen Sie auf der Seite "Permissions" (Berechtigungen) die Option Edit (Bearbeiten) aus.
4. Aktivieren Sie in der Benutzerliste das Kontrollkästchen für SSH/RDP für den Benutzer, dem Sie die erforderlichen Berechtigungen erteilen möchten. Wenn Sie dem Benutzer außerdem auch Administratorberechtigungen gewähren möchten, wählen Sie noch sudo/admin (sudo/admin) aus.



5. Wählen Sie Speichern.

Jetzt kann der Benutzer ein Passwort erhalten und sich damit wie nachfolgend beschrieben bei der Instance anmelden.

> **Note**
>
> Sie können sich auch als Administrator anmelden. Weitere Informationen finden Sie unter Anmelden als Administrator.

Schritt 2.3: Implementieren eines benutzerdefinierten Rezeptbuchs

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Obwohl ein Stack im Grunde ein Container für Instances ist, fügen Sie einem Stack Instances nicht direkt hinzu. Sie fügen mindestens einen Layer hinzu. Jeder Layer repräsentiert dabei eine Gruppe zusammengehöriger Instances, die Sie dann den Layers hinzufügen.

Eine Ebene ist im Grunde ein Blueprint, den AWS OpsWorks Stacks verwendet, um eine Reihe von Amazon EC2 EC2-Instances mit derselben Konfiguration zu erstellen. Eine Instance beginnt mit einer Basisversion des Betriebssystems und der Layer der Instance führt verschiedene Aufgaben auf der Instance aus, um dieses Konzept zu implementieren, darunter:

- Erstellen von Verzeichnissen und Dateien
- Verwalten von Benutzern
- Installieren und Konfigurieren von Software
- Starten oder Beenden von Servern
- Bereitstellen von Anwendungscode und zugehörigen Dateien

Eine Ebene führt Aufgaben auf Instances durch, indem sie Chef-Rezepte — kurz Rezepte — ausführt. Ein Rezept ist eine Ruby-Anwendung, die mithilfe der domänenspezifischen Sprache von Chef den endgültigen Zustand der Instance beschreibt. Bei AWS OpsWorks Stacks wird jedes Rezept normalerweise einem der Lebenszyklusereignisse der Ebene zugewiesen: Setup, Configuration, Deploy, Undeploy und Shutdown. Wenn ein Lebenszyklusereignis auf einer Instanz eintritt, führt AWS OpsWorks Stacks die Rezepte des Ereignisses aus, um die entsprechenden Aufgaben auszuführen. Das Setup-Ereignis tritt beispielsweise ein, nachdem der Startvorgang einer Instanz abgeschlossen ist. AWS OpsWorks Stacks führt dann die Setup-Rezepte aus, mit denen normalerweise Aufgaben wie das Installieren und Konfigurieren von Serversoftware und das Starten der zugehörigen Dienste ausgeführt werden.

AWS OpsWorks Stacks stellt jeder Ebene eine Reihe von integrierten Rezepten zur Verfügung, mit denen Standardaufgaben ausgeführt werden. Mithilfe von benutzerdefinierten Rezepten, die Sie den einzelnen Lebenszyklusereignissen eines Layers zuweisen, können Sie die Funktionalität eines Layers um zusätzliche Aufgaben erweitern. Windows-Stacks unterstützen benutzerdefinierte Ebenen mit einem minimalen Rezeptsatz ausschließlich für einige Basisaufgaben. Um Ihren Windows-Instances Funktionen hinzuzufügen, müssen Sie benutzerdefinierte Rezepte implementieren, um

beispielsweise Software zu installieren oder Anwendungen bereitzustellen. In diesem Thema wird beschrieben, wie Sie einen einzelnen benutzerdefinierten Layer erstellen, um IIS-Instances zu unterstützen.

Themen

- Eine kurze Einführung in Rezeptbücher und Rezepte
- Implementieren eines Rezepts zum Installieren und Starten von IIS
- Aktivieren des benutzerdefinierten Rezeptbuchs

Eine kurze Einführung in Rezeptbücher und Rezepte

Ein Rezept definiert mindestens einen Aspekt des erwarteten Status einer Instance: welche Verzeichnisse sie haben sollte, welche Softwarepakete installiert sein sollten, welche Apps bereitgestellt werden sollten usw. Rezepte sind in einem Rezeptbuch zusammengefasst. Dieses enthält in der Regel ein oder mehrere zusammengehörige Rezepte sowie die zugehörigen Dateien wie Vorlagen zum Erstellen von Konfigurationsdateien.

In diesem Thema werden Sie grundlegend an Rezepte herangeführt und erfahren, wie Sie ein Rezeptbuch implementieren, um einen einfachen benutzerdefinierten IIS-Layer zu unterstützen. Weitere allgemeine Informationen zu Rezeptbüchern finden Sie unter Cookbooks und Rezepte. Eine detaillierte Einführung in die Implementierung von Rezeptbüchern einschließlich Windows-spezifischer Themen finden Sie unter Rezeptbücher 101.

Chef-Rezepte sind technisch gesehen Ruby-Anwendungen. Ein Großteil des Codes ist jedoch in Chef DSL geschrieben. DSL besteht größtenteils aus einer Reihe von Ressourcen, mithilfe derer sich Aspekte des Zustands einer Instance beschreiben lassen. Eine `directory`-Ressource definiert beispielsweise ein Verzeichnis, das einem System hinzugefügt werden soll. Im folgenden Beispiel wird das Verzeichnis `C:\data` des angegebenen Benutzers mit umfassenden Rechten definiert, das keine Rechte vom übergeordneten Verzeichnis erbt.

```
directory 'C:\data' do
  rights :full_control, 'WORKGROUP\username'
  inherits false
  action :create
end
```

Wenn Chef ein Rezept ausführt, wird jede Ressource einzeln ausgeführt. Dabei werden die Daten an einen zugehörigen Anbieter übergeben, ein Ruby-Objekt, das die Einzelheiten bei der Modifizierung

des Instance-Zustands verarbeitet. In diesem Fall erstellt der Anbieter ein neues Verzeichnis mit der angegebenen Konfiguration.

Das benutzerdefinierte Rezeptbuch für den benutzerdefinierten IIS-Layer muss die folgenden Aufgaben ausführen:

- Installieren der IIS-Funktion und Starten des Service

  Diese Aufgabe wird in der Regel während der Einrichtung direkt nach dem Hochfahren der Instance ausgeführt.
- Bereitstellen einer App für die Instance, in diesem Beispiel eine einfache HTML-Seite

  Diese Aufgabe wird in der Regel während der Einrichtung ausgeführt. Apps müssen üblicherweise jedoch regelmäßig aktualisiert werden, daher müssen Sie auch Updates bereitstellen, während die Instance bereits online ist.

Sie können alle diese Aufgaben mit nur einem Rezept ausführen. Es empfiehlt sich jedoch, für Einrichtung und Bereitstellung jeweils eigene Rezepte zu verwenden. So können Sie App-Updates jederzeit bereitstellen, ohne dafür Einrichtungscode ausführen zu müssen. Nachfolgend wird beschrieben, wie Sie ein Rezeptbuch einrichten, um einen benutzerdefinierten IIS-Layer zu unterstützen. In den nachfolgenden Themen erfahren Sie, wie Sie die Rezepte implementieren.

Dies sind Ihre ersten Schritte

1. Erstellen Sie ein Verzeichnis namens `iis-cookbook` in einem lokalen Verzeichnis auf Ihrem Computer.
2. Fügen Sie eine Datei `metadata.rb` mit dem folgenden Inhalt zu `iis-cookbook` hinzu.

   ```
   name "iis-cookbook"
   version "0.1.0"
   ```

   In diesem Beispiel enthält die Datei `metadata.rb` nur minimale Daten. Weitere Informationen zur Verwendung dieser Datei finden Sie unter metadata.rb.
3. Fügen Sie ein Verzeichnis `recipes` zu `iis-cookbook` hinzu.

   Dieses Verzeichnis, das den Namen `recipes` haben muss, enthält die Rezepte des Rezeptbuchs.

Im Allgemeinen können Rezeptbücher verschiedene andere Verzeichnisse enthalten. Wenn ein Rezept beispielsweise eine Vorlage zum Erstellen einer Konfigurationsdatei enthält, wird diese Vorlage normalerweise im Verzeichnis `templates\default` gespeichert. Das Rezeptbuch in diesem Beispiel besteht nur aus Rezepten und benötigt daher keine weiteren Verzeichnisse. In diesem Beispiel wird außerdem nur ein einziges Rezeptbuch verwendet. Für komplexere Projekte können Sie jedoch auch beliebig viele Rezeptbücher verwenden. Es bietet sich beispielsweise an, für Einrichtung und Bereitstellung jeweils eigene Rezeptbücher zu verwenden. Weitere Beispiele für Rezeptbücher finden Sie unter Cookbooks und Rezepte.

Implementieren eines Rezepts zum Installieren und Starten von IIS

IIS ist eine Windows-Funktion, die zu einer Reihe optionaler Systemkomponenten gehört, die Sie auf Windows Server installieren können. Sie können IIS mithilfe eines Rezepts auf eine der folgenden Weisen installieren:

- Verwenden Sie eine `powershell_script`-Ressource, um das `Install-WindowsFeature`-Cmdlet auszuführen.
- Verwenden Sie das Chef-Windows-Rezeptbuch`windows_feature`.

  Das `windows`-Rezeptbuch enthält eine Reihe von Ressourcen, deren Anbieter mithilfe von Abbildbereitstellung und Verwaltung (Deployment Image Servicing and Management, DISM)) verschiedene Aufgaben wie die Installation von Funktionen auf Windows-Instances ausführen.

> **Note**
>
> `powershell_script` ist eine der wichtigsten Ressourcen für Windows-Rezepte. Sie können damit eine Vielzahl von Aufgaben auf einer Instanz ausführen, indem Sie ein PowerShell Skript oder Cmdlet ausführen. Skripte sind insbesondere für Aufgaben hilfreich, die von Chef-Ressourcen nicht unterstützt werden.

In diesem Beispiel wird ein PowerShell Skript zum Installieren und Starten des Webservers (IIS) ausgeführt. Das `windows`-Rezeptbuch wird im weiteren Verlauf dieser Anleitung beschrieben. Ein Beispiel für die Installation von IIS mithilfe von `windows_feature` finden Sie unter Installieren einer Windows-Funktion: IIS.

Fügen Sie ein Rezept namens `install.rb` mit dem folgenden Inhalt zum Verzeichnis `recipes` des Rezeptbuchs hinzu.

```
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
```

Das Rezept enthält zwei Ressourcen.

powershell_script

`powershell_script`führt das angegebene PowerShell Skript oder Cmdlet aus. Das Beispiel verwendet folgende Attributeinstellungen:

- `code`— Die auszuführenden PowerShell Cmdlets.

  In diesem Beispiel wird ein `Install-WindowsFeature`-Cmdlet zur Installation von Web Server (IIS) ausgeführt. Allgemein kann das Attribut `code` beliebig viele Zeilen haben. Sie können also alle Cmdlets ausführen, die Sie benötigen.
- `not-if`— Ein Guard-Attribut, das sicherstellt, dass das Rezept IIS nur installiert, wenn es noch nicht installiert wurde.

  Rezepte sind in der Regel idempotent, damit sie dieselbe Aufgabe nur einmal ausführen.

Jede Ressource hat eine Aktion, über die die Aktion festgelegt ist, die der Anbieter ausführt. Für dieses Beispiel gibt es keine explizite Aktion, daher führt der Anbieter die `:run` Standardaktion aus, bei der das angegebene PowerShell Skript ausgeführt wird. Weitere Informationen finden Sie unter Ein PowerShell Windows-Skript ausführen.

Service nicht zulässig

Ein `service` verwaltet einen Service, in diesem Fall den Web Server IIS-Service (W3SVC). Im Beispiel werden Standardattribute verwendet und zwei Aktionen festgelegt, `:start` und `:enable`, um IIS zu starten und zu aktivieren.

> **Note**
>
> Wenn Sie Software über ein Paketinstallationsprogramm wie MSI installieren möchten, können Sie eine `windows_package`-Ressource verwenden. Weitere Informationen finden Sie unter Installieren eines Pakets.

Aktivieren des benutzerdefinierten Rezeptbuchs

AWS OpsWorks Stacks führt auf jeder Instanz Rezepte aus einem lokalen Cache aus. Gehen Sie wie folgt vor, um eigene benutzerdefinierte Rezepte auszuführen:

- Speichern Sie das Rezeptbuch in einem Remote-Repository.

  AWS OpsWorks Stacks lädt die Kochbücher aus diesem Repository in den lokalen Cache jeder Instanz herunter.
- Bearbeiten Sie den Stack, um benutzerdefinierte Rezeptbücher zu aktivieren.

  Benutzerdefinierte Rezeptbücher sind standardmäßig deaktiviert und müssen für den Stack zunächst aktiviert werden. Außerdem müssen Sie die URL des Repositorys sowie die dazugehörigen Informationen angeben.

AWS OpsWorks Stacks unterstützt S3-Archive und Git-Repositorys für benutzerdefinierte Kochbücher. In diesem Beispiel wird ein S3-Archiv verwendet. Weitere Informationen finden Sie unter Rezeptbuch-Repositorys.

So verwenden Sie ein S3-Archiv

1. Erstellen Sie ein `.zip`-Archiv des Verzeichnisses `iis-cookbook`.

   AWS OpsWorks Stacks unterstützt auch `.tgz` (mit Gzip komprimierte Tar-) Archive für Windows-Stacks.
2. Laden Sie das Archiv in einen S3-Bucket in der Region USA West (Nordkalifornien) hoch und veröffentlichen Sie die Datei. Sie können auch private S3-Archive verwenden, öffentliche Archive sind für dieses Beispiel jedoch ausreichend und einfacher zu handhaben.

   a. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/s3/.

b. Wenn Sie noch keinen Bucket angelegt habenus-west-1, wählen Sie Create Bucket und erstellen Sie einen Bucket in der Region USA West (Nordkalifornien).

c. Wählen Sie in der Bucket-Liste den Namen des Buckets aus, in den Sie die Datei hochladen möchten, und danach Upload (Hochladen).

d. Wählen Sie Add Files (Dateien hinzufügen) aus.

e. Wählen Sie die hochzuladende Archivdatei und danach Open (Öffnen).

f. Wählen Sie unten im Dialog Upload - Select Files and Folders (Hochladen - Dateien und Ordner auswählen) die Option Set Details (Details festlegen) aus.

g. Wählen Sie unten im DialogSet Details (Details festlegen) die Option Set Permissions (Berechtigungen festlegen) aus.

h. Wählen Sie im Dialog Set Permissions (Berechtigungen festlegen) die Option Make everything public (Alles veröffentlichen) aus.

i. Wählen Sie unten im DialogSet Permissions (Berechtigungen festlegen) die Option Start Upload (Hochladen starten) aus. Nach dem Upload wird die Datei `iis-cookbook.zip` in Ihrem Bucket angezeigt.

j. Wählen Sie den Bucket und danach die Registerkarte Properties (Eigenschaften) für den Bucket aus. Notieren Sie sich neben Link (Link) die URL der Archivdatei.

Weitere Informationen zum Hochladen von Dateien in einen Amazon S3 S3-Bucket finden Sie unter Wie lade ich Dateien und Ordner in einen S3-Bucket hoch? im Amazon S3 S3-Konsolen-Benutzerhandbuch.

> **Important**
>
> Bisher hat diese Anleitung Sie nur wenig Zeit gekostet und AWS OpsWorks Stacks selbst ist kostenlos. Sie müssen jedoch für alle AWS-Ressourcen bezahlen, die Sie verwenden, z. B. Amazon S3 S3-Speicher. Sobald Sie das Archiv hochladen, entstehen Ihnen Kosten. Weitere Informationen finden Sie unter AWS-Preise.

So aktivieren Sie benutzerdefinierte Rezeptbücher für den Stack

1. Wählen Sie in der AWS OpsWorks Stacks-Konsole im Navigationsbereich Stack und dann oben rechts Stack-Einstellungen aus.

2. Wählen Sie oben rechts auf der Seite Settings (Einstellungen) die Option Edit (Bearbeiten) aus.
3. Legen Sie auf der Seite Settings (Einstellungen) die Option Use custom Chef cookbooks (Benutzerdefinierte Chef-Rezeptbücher verwenden) auf Yes (Ja) fest und geben Sie die folgenden Informationen ein:
   - Repository-Typ — S3-Archiv.
   - Repository-URL — Die S3-URL der Kochbuch-Archivdatei, die Sie zuvor aufgenommen haben.
4. Wählen Sie Save (Speichern) aus, um die Stack-Konfiguration zu aktualisieren.

AWS OpsWorks Stacks installiert dein benutzerdefiniertes Kochbuch auf allen neuen Instanzen. Auf Online-Instances werden benutzerdefinierte Rezeptbücher jedoch nicht automatisch durch AWS OpsWorks Stacks installiert oder aktualisiert. Dies können Sie, wie im weiteren Verlauf dieser Anleitung beschrieben, manuell tun.

Schritt 2.4: Hinzufügen eines IIS-Layers

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Ihr Rezeptbuch enthält ein Rezept, mit dem IIS installiert und gestartet wird. Dies ist ausreichend, um den Layer zu erstellen und sicherzustellen, dass die IIS-Instance ordnungsgemäß funktioniert. Später werden Sie eine Funktion zur Anwendungsbereitstellung zum Layer hinzufügen.

Erstellen eines Layers

Fügen Sie dem Stack zunächst einen Layer hinzu. Dann fügen Sie dem Layer Funktionen hinzu, indem Sie den entsprechenden Lebenszyklusereignissen benutzerdefinierte Rezepte hinzufügen.

So fügen Sie einem Stack einen IIS-Layer hinzu

1. Wählen Sie im Navigationsbereich Layers (Layer) und dann Add a layer (Layer hinzufügen) aus.

2. Konfigurieren Sie den Layer wie folgt:

   - Name — **`IISExample`**
   - Kurzname — **`iisexample`**

     AWS OpsWorks Stacks verwendet den Kurznamen, um die Ebene intern zu identifizieren. Außerdem wird der kurze Name verwendet, um den Layer in Rezepten zu identifizieren (nicht in diesem Beispiel). Sie können einen beliebigen kurzen Namen angeben. Er darf jedoch nur Kleinbuchstaben und eine geringe Anzahl an Satzzeichen enthalten. Weitere Informationen finden Sie unter Benutzerspezifische Layers.

3. Wählen Sie Add Layer (Ebene hinzufügen) aus.

Wenn Sie jetzt IISWalkthrough eine Instance hinzufügen und diese starten würden, würde AWS OpsWorks Stacks die Rezeptbücher automatisch installieren. `install.rb` würde jedoch nicht ausgeführt werden. Sobald eine Instance online ist, können Sie Rezepte mit dem Stack-Befehl "execute recipes" manuell ausführen. Ein besserer Ansatz besteht jedoch darin, das Rezept einem der Lebenszyklusereignisse der Ebene zuzuweisen. AWS OpsWorks Stacks führt das Rezept dann automatisch an der entsprechenden Stelle im Lebenszyklus der Instanz aus.

Installieren und starten Sie IIS, sobald die Instance gestartet wurde. Zu diesem Zweck weisen Sie dem `Setup`-Ereignis des Layers `install.rb` zu.

So weisen Sie das Rezept einem Lebenszyklusereignis zu

1. Wählen Sie im Navigationsbereich Layers (Ebenen) aus.
2. Wählen Sie im Feld für die Ebene IISExample die Option Recipes (Rezepte) aus.
3. Wählen Sie rechts oben Edit (Bearbeiten) aus.
4. Geben Sie unter Custom Chef Recipes (Benuterdefinierte Chef-Rezepte) im Rezeptfeld Setup (Einrichten) die Option **`iis-cookbook::install`** ein.

   > **Note**
   >
   > Rezepte werden anhand von *`cookbook-name`*`::`*`recipe-name`* identifiziert, wobei das Suffix `.rb` des Rezeptnamens weggelassen wird.

5. Wählen Sie das +-Symbol aus, um das Rezept der Ebene hinzuzufügen. Neben dem Rezept wird ein rotes "X" angezeigt, über das Sie es später einfach entfernen können.

6. Wählen Sie Save (Speichern) aus, um die neue Konfiguration zu speichern. Die benutzerdefinierten Einrichtungsrezepte sollten nun `iis-cookbook::install` enthalten.

Hinzufügen einer Instance zum Layer und Starten der Instance

Sie können das Rezept ausprobieren, indem Sie der Ebene eine Instanz hinzufügen und die Instanz starten. AWS OpsWorks Stacks installiert die Kochbücher automatisch und wird `install.rb` während des Setups ausgeführt, sobald die Instanz mit dem Booten fertig ist.

So fügen Sie einem Layer eine Instance hinzu und starten diese

1. Wählen Sie im AWS OpsWorks Stacks-Navigationsbereich die Option Instances aus.
2. Wählen Sie unter der Ebene IISExample die Option Add an instance (Instance hinzufügen) aus.
3. Wählen Sie die entsprechende Größe aus. t2.micro (t2.micro) (oder die kleinste verfügbare Größe) sollte für das Beispiel ausreichen.
4. Wählen Sie Add Instance (Instance hinzufügen) aus. Standardmäßig generiert AWS OpsWorks Stacks Instanznamen, indem eine Ganzzahl an den Kurznamen der Ebene angehängt wird. Daher sollte die Instanz den Namen iisexample1 haben.
5. Wählen Sie Start in der Spalte Aktionen der Instanz aus, um die Instanz zu starten. AWS OpsWorks Stacks startet dann eine EC2-Instance und führt die Setup-Rezepte aus, um sie zu konfigurieren. Wenn der Layer zu diesem Zeitpunkt über Deploy-Rezepte verfügte, würde AWS OpsWorks Stacks diese ausführen, nachdem die Setup-Rezepte abgeschlossen sind.

   Dies kann einige Minuten dauern. Der Status in der Spalte Status (Status) wechselt in dieser Zeit mehrfach. Sobald der Status online (Online) angezeigt wird, ist der Einrichtungsvorgang abgeschlossen und die Instance kann verwendet werden.

Bestätigen, dass IIS installiert ist und ausgeführt wird

Sie können sich über RDP bei der Instance anmelden und überprüfen, ob das Einrichtungsrezept korrekt ausgeführt wurde.

So überprüfen Sie, dass IIS installiert ist und ausgeführt wird

1. Wählen Sie im Navigationsbereich Instances und in der Spalte Actions der iisexample1-Instanz die Option rdp aus. AWS OpsWorks Stacks generiert automatisch ein RDP-Passwort für Sie, das nach einem bestimmten Zeitraum abläuft.

2. Legen Sie Session valid for (Sitzung gültig für) auf 2 Stunden fest und wählen Sie Generate Password (Passwort generieren) aus.
3. AWS OpsWorks Stacks zeigt das Passwort und der Einfachheit halber auch den öffentlichen DNS-Namen und den Benutzernamen der Instanz an. Kopieren Sie alle drei und klicken Sie auf Acknowledge and close (Bestätigen und schließen).
4. Öffnen Sie den RDP-Client und verwenden Sie die Daten aus Schritt 3, um eine Verbindung zur Instance herzustellen.
5. Öffnen Sie den Windows Explorer in der Instance und untersuchen Sie das Laufwerk `C:`. Hier sollte während der Installation von IIS ein Verzeichnis namens `C:\inetpub` angelegt worden sein.
6. Öffnen Sie in der Systemsteuerung die Anwendung Verwaltung und dann Dienste. Der IIS-Service sollte unten in der Liste angezeigt werden. Er hat den Namen "World Wide Web Publishing Service" und sollte den Status Wird ausgeführt haben.
7. Kehren Sie zur AWS OpsWorks Stacks-Konsole zurück und wählen Sie die öffentliche IP-Adresse der iisexample1-Instanz aus. Stellen Sie sicher, dass Sie dies in AWS OpsWorks Stacks und nicht in der Amazon EC2 EC2-Konsole tun. Hierüber wird automatisch eine HTTP-Anfrage an die Adresse gesendet. Es sollte sich die Standard-IIS-Willkommensseite öffnen.

Im nächsten Thema wird erläutert, wie eine App auf der Instance bereitgestellt wird. In diesem Beispiel handelt es sich dabei um eine einfache statische HTML-Seite. Wenn Sie allerdings lieber eine Pause machen möchten, wählen Sie in der Spalte Actions (Aktionen) der Instance iisexample1 die Option stop (Anhalten) aus, um die Instance anzuhalten und unnötige Gebühren zu vermeiden. Sie können die Instance jederzeit neu starten, wenn Sie bereit sind fortzufahren.

Schritt 2.5: Bereitstellen einer App

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Bei der IIS-Installation wird ein Verzeichnis `C:\inetpub\wwwroot` für den Anwendungscode und die zugehörigen Dateien erstellt. Als nächstes installieren Sie eine App in diesem Verzeichnis. Für dieses Beispiel werden Sie eine statische HTML-Homepage, `default.html`, in `C:\inetpub\wwwroot` installieren. Dieser allgemeine Ansatz lässt sich einfach auf komplexere Szenarios wie ASP.NET-Anwendungen erweitern.

Sie können die Anwendungsdateien in Ihr Rezeptbuch aufnehmen und sie über `install.rb` in `C:\inetpub\wwwroot` kopieren. Beispiele für diese Vorgehensweise finden Sie unter Beispiel 6: Erstellen von Dateien. Dieser Ansatz ist allerdings nicht sonderlich flexibel oder effizient. Es empfiehlt sich daher in der Regel, die Bereitstellung von Rezeptbüchern und Anwendungen zu trennen.

Die bevorzugte Lösung besteht darin, ein separates Bereitstellungsrezept zu implementieren, das den Code der Anwendung und die zugehörigen Dateien aus einem Repository — einem beliebigen Repository, nicht nur dem Cookbook-Repository — abruft und es auf jeder IIS-Serverinstanz installiert. So wird die Bereitstellung von Rezeptbüchern und Anwendungen sauber getrennt, sodass Sie beim Aktualisieren von Anwendungen das Bereitstellungsrezept erneut ausführen können, ohne das Rezeptbuch aktualisieren zu müssen.

In diesem Thema wird gezeigt, wie Sie ein einfaches Bereitstellungsrezept implementieren, über das `default.htm` auf Ihrem IIS-Server bereitgestellt wird. Sie können diese Beispiel einfach auf komplexere Anwendungen übertragen.

Themen

- Erstellen der Anwendung und Speichern in einem Repository
- Implementieren eines Rezepts für die Bereitstellung der Anwendung
- Aktualisieren der Rezeptbücher der Instance
- Hinzufügen des Rezepts zum benutzerdefinierten IIS-Layer
- Hinzufügen einer Anwendung
- Bereitstellen der App und Öffnen der Anwendung

Erstellen der Anwendung und Speichern in einem Repository

Sie können für Ihre Anwendungen ein beliebiges Repository verwenden. Der Einfachheit halber wird `default.htm` in diesem Beispiel in einem öffentlichen S3-Bucket gespeichert.

So erstellen Sie die Anwendung

1. Erstellen Sie ein Verzeichnis namens `iis-application` in einem lokalen Verzeichnis auf Ihrem Computer.
2. Fügen Sie eine Datei `default.htm` zu `iis-application` mit dem folgenden Inhalt hinzu:

```
<!DOCTYPE html>
<html>
  <head>
    <title>IIS Example</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

3. Erstellen Sie einen S3-Bucket, laden Sie `default.htm` auf den Bucket hoch und notieren Sie sich die URL. Machen Sie die Datei der Einfachheit halber öffentlich.

> **Note**
>
> Dies ist eine äußerst einfache Anwendung, die Grundprinzipien lassen sich jedoch einfach auf Anwendungen für die Produktion ausweiten.
>
> - Bei komplexeren Anwendungen mit mehreren Dateien ist es in der Regel einfacher, ein ZIP-Archiv von `iis-application` zu erstellen und dieses auf Ihren S3-Bucket hochzuladen.
>
>   Diese ZIP-Datei können Sie dann herunterladen und den Inhalt in das entsprechende Verzeichnis extrahieren. So müssen Sie weder mehrere Dateien herunterladen noch eine Verzeichnisstruktur nachbilden.
> - Bei Produktionsanwendungen sollten Sie Ihre Dateien privat halten. Ein Beispiel, wie Sie mit einem Rezept Dateien aus einem privaten S3-Bucket herunterladen, finden Sie unter Verwenden des SDK for Ruby auf einer AWS OpsWorks Stacks-Windows-Instanz.
> - Sie können Anwendungen in jedem geeigneten Repository speichern.
>
>   Anwendungen werden üblicherweise über die öffentliche API eines Repositorys heruntergeladen. In diesem Beispiel wird die Amazon S3 S3-API verwendet. Wenn

> Sie Ihre Anwendung beispielsweise auf speichern GitHub, können Sie die GitHub API verwenden.

Implementieren eines Rezepts für die Bereitstellung der Anwendung

Fügen Sie ein Rezept namens `deploy.rb` dem Verzeichnis `iis-cookbook` `recipes` hinzu, mit folgendem Inhalt.

```
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
```

In diesem Beispiel wird SDK for Ruby v2 verwendet, um die Datei herunterzuladen. AWS OpsWorks Stacks installiert dieses SDK jedoch nicht auf Windows-Instanzen, sodass das Rezept mit der `chef_gem`Ressource beginnt, die diese Aufgabe erledigt.

> **Note**
>
> Die Ressource `chef_gem` installiert Gems in der Chef-eigenen Ruby-Version. Dies ist die Version, die in Rezepten verwendet wird. Wenn Sie ein Gem für eine systemweite Ruby-Version installieren möchten, verwenden Sie die Ressource gem_package.

Der Großteil des Rezepts ist eine `ruby_block`Ressource, die einen Block von Ruby-Code ausführt, der das SDK for Ruby zum Herunterladen verwendet`default.htm`. Der Code im `ruby_block` lässt sich in folgende Bereiche unterteilen, die den durchnummerierten Kommentaren im Codebeispiel entsprechen.

1: Angeben eines Zertifikat-Bundles

Amazon S3 verwendet SSL, sodass Sie ein entsprechendes Zertifikat benötigen, um Objekte aus einem S3-Bucket herunterzuladen. Das SDK for Ruby v2 enthält kein Zertifikatspaket, daher müssen Sie eines bereitstellen und das SDK for Ruby konfigurieren, um es verwenden zu können. AWS OpsWorks Stacks installiert ein Zertifikatspaket nicht direkt, aber es installiert Git, das ein Zertifikatspaket (`curl-ca-bundle.crt`) enthält. Der Einfachheit halber konfiguriert dieses Beispiel das SDK for Ruby so, dass es das Git-Zertifikatspaket für SSL verwendet. Sie können auch eigene Bundles installieren und das SDK entsprechend konfigurieren.

2: Abrufen der Repository-Daten

Um ein Objekt von Amazon S3 herunterzuladen, benötigen Sie die AWS-Region, den Bucket-Namen und den Schlüsselnamen. Wie nachfolgend beschrieben werden in diesem Beispiel diese Informationen bereitgestellt, indem eine Reihe von Umgebungsvariablen der Anwendung zugeordnet werden. Wenn Sie eine App bereitstellen, fügt AWS OpsWorks Stacks dem Knotenobjekt der Instance eine Reihe von Attributen hinzu. Diese Attribute bestehen aus einer Hash-Tabelle, die die Anwendungskonfiguration einschließlich der Umgebungsvariablen enthält. Die Anwendungsattribute für diese Anwendung im JSON-Format sehen etwa wie folgt aus.

```
{
  "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
}
```

Die Umgebungsvariablen der Anwendung werden im Attribut `[:environment]` gespeichert. Um diese abzurufen, verwenden Sie eine Chef-Suchanfrage, mit der Sie die Hash-Tabelle der Anwendung abrufen, die im Knoten `aws_opsworks_app` gespeichert ist. Diese Anwendung hat den Typ `other`, die Anfrage sucht also nach Anwendungen dieses Typs. Das Rezept nutzt die Tatsache, dass sich auf dieser Instance nur eine Anwendung befindet. Die gewünschte Hash-Tabelle ist also einfach `app[0]`. Der Einfachheit halber weist das Rezept dann die Region-, Bucket- und Dateinamen Variablen zu.

Weitere Informationen zur Verwendung der Chef-Suche finden Sie unter Abrufen von Attributwerten mit der Chef-Suche.

3: Herunterladen der Datei

Im dritten Teil des Rezepts wird ein S3-Client-Objekt erstellt und mit der Methode `get_object` die Datei `default.htm` in das Verzeichnis `C:\inetpub\wwwroot` der Instance heruntergeladen.

> **Note**
>
> Ein Rezept ist eine Ruby-Anwendung, der Ruby-Code muss sich also nicht unbedingt in einem `ruby_block` befinden. Der Code im Text des Rezepts wird jedoch zuerst ausgeführt. Danach werden die Ressourcen der Reihe nach abgearbeitet. Wenn Sie in diesem Beispiel den Download-Code in den Rezepttext einfügen, schlägt dies fehl, da die `chef_gem` Ressource das SDK for Ruby noch nicht installiert hätte. Der Code in der `ruby_block` Ressource wird ausgeführt, wenn die Ressource ausgeführt wird, nachdem die `chef_gem` Ressource das SDK for Ruby installiert hat.

Aktualisieren der Rezeptbücher der Instance

AWS OpsWorks Stacks installiert automatisch benutzerdefinierte Kochbücher auf neuen Instanzen. Sie arbeiten jedoch mit einer bestehenden Instance und müssen Ihr Rezeptbuch daher manuell aktualisieren.

So aktualisieren Sie die Rezeptbücher der Instance

1. Erstellen Sie ein `.zip`-Archiv von `iis-cookbook` und laden Sie es in den S3-Bucket hoch.

   Das vorhandene Rezeptbuch wird dadurch überschrieben. Die URL bleibt jedoch dieselbe, Sie müssen die Stack-Konfiguration also nicht aktualisieren.
2. Wenn Ihre Instance nicht online ist, starten Sie sie neu.
3. Wenn die Instance online ist, wählen Sie im Navigationsbereich Stack und dann Run Command (Befehl ausführen) aus.
4. Wählen Sie für Command (Befehl) Update Custom Cookbooks (Benutzerdefinierte Rezeptbücher aktualisieren) aus. Mit diesem Befehl installieren Sie das aktualisierte Rezeptbuch auf der Instance.
5. Wählen Sie Update Custom Cookbooks (Benutzerdefinierte Rezeptbücher aktualisieren) aus. Die Ausführung dieses Befehls kann einige Minuten dauern.

Hinzufügen des Rezepts zum benutzerdefinierten IIS-Layer

Wie auch bei `install.rb` ist die bevorzugte Methode für die Bereitstellung, `deploy.rb` dem entsprechenden Lebenszyklusereignis zuzuweisen. Rezepte zur Bereitstellung werden normalerweise dem Ereignis "Bereitstellung" zugewiesen und zusammenfassend als Bereitstellungsrezepte bezeichnet. Allein durch die Zuweisung zum Ereignis "Bereitstellung" wird das Ereignis noch nicht ausgelöst. Stattdessen geschieht Folgendes:

- Bei neuen Instanzen führt AWS OpsWorks Stacks die Deploy-Rezepte automatisch aus, nachdem die Setup-Rezepte abgeschlossen sind, sodass neue Instanzen automatisch über die aktuelle Anwendungsversion verfügen.
- Bei Online-Instances verwenden Sie einen Bereitstellungsbefehl, um neue oder aktualisierte Anwendungen manuell zu installieren.

  Über diesen Befehl wird ein Bereitstellungsereignis auf den Instances des Stack ausgelöst, das wiederum die Bereitstellungsrezepte ausführt.

So weisen Sie deploy.rb dem Ereignis "Bereitstellung" des Layers zu

1. Wählen Sie im Navigationsbereich Layers (Ebenen) und dann Recipes (Rezepte) unter Layer IISExample (Ebene IISExample) aus.
2. Fügen Sie unter Custom Chef Recipes (Benutzerdefinierte Chef-Rezepte) die Zeichenfolge **`iis-cookbook::deploy`** dem Rezeptfeld Deploy (Bereitstellen) hinzu und wählen Sie + aus, um der Ebene das Rezept hinzuzufügen.
3. Wählen Sie Save (Speichern) aus, um die neue Konfiguration zu speichern. Die benutzerdefinierten Bereitstellungsrezepte sollten nun `iis-cookbook::deploy` enthalten.

Hinzufügen einer Anwendung

Die letzte Aufgabe besteht darin, dem Stack eine App hinzuzufügen, die Ihre Anwendung in der AWS OpsWorks Stacks-Umgebung repräsentiert. Eine App enthält Metadaten wie den Anzeigenamen der Anwendung sowie die Daten, die Sie zum Herunterladen der App aus dem Repository benötigen.

So fügen Sie dem Stack die App hinzu

1. Wählen Sie im Navigationsbereich Apps (Anwendungen) und dann Add an app (Anwendung hinzufügen) aus.
2. Konfigurieren Sie die App mit den folgenden Einstellungen.

- Name — ich **`IIS-Example-App`**
- Repository-Typ — Andere
- Umgebungsvariablen — Fügen Sie die folgenden drei Umgebungsvariablen hinzu:
  - **`S3REGION`**— Die Region des Buckets (in diesem Fall`us-west-1`).
  - **`BUCKET`**— Der Bucket-Name, z. `windows-example-app` B.
  - **`FILENAME`**— Der Dateiname:**`default.htm`**.

3. Akzeptieren Sie Standardwerte für die übrigen Einstellungen und wählen Sie dann Add App (Anwendung hinzufügen) aus, um dem Stack die Anwendung hinzuzufügen.

> **Note**
>
> In diesem Beispiel werden die Download-Daten über Umgebungsvariablen bereitgestellt. Ein alternativer Ansatz besteht darin, einen S3-Archive-Repository-Typ zu verwenden und die URL der Datei anzugeben. AWS OpsWorks Stacks fügt die Informationen zusammen mit optionalen Daten wie Ihren AWS-Anmeldeinformationen zum `app_source` Attribut der App hinzu. Ihr Bereitstellungsrezept muss daraufhin die URL aus den App-Attributen abrufen und daraus die Region, den Bucket-Namen und den Dateinamen extrahieren.

Bereitstellen der App und Öffnen der Anwendung

AWS OpsWorks Stacks stellt Apps automatisch für neue Instances bereit, nicht jedoch für Online-Instances. Da Ihre Instance bereits online ist, müssen Sie die App manuell bereitstellen.

So stellen Sie die App bereit

1. Wählen Sie im Navigationsbereich Apps (Anwendungen) und dann in der Spalte Actions (Aktionen) der Anwendung deploy (Bereitstellen) aus.
2. Command (Befehl) muss auf Deploy (Bereitstellen) eingestellt sein. Wählen Sie Deploy (Bereitstellen) rechts unten auf der Seite Deploy App (App bereitstellen). Die Ausführung dieses Befehls kann einige Minuten dauern.

   Nachdem die Bereitstellung abgeschlossen wurde, kehren Sie zur Seite Apps zurück. Die Anzeige Status zeigt in grüner Schrift successful (Erfolgreich) an und neben dem Namen der Anwendung wird die erfolgreiche Bereitstellung durch ein grünes Häkchen dargestellt.

> **Note**
>
> Windows-Apps haben grundsätzlich den App-Typ Other (Sonstige). Durch die Bereitstellung der App geschieht daher Folgendes:
>
> - Die Anwendungsdaten werden wie zuvor beschrieben den Stack-Konfigurations- und Bereitstellungsattributen hinzugefügt.
> - Es wird ein Bereitstellungsereignis auf den Instances des Stacks ausgelöst, durch das Ihre benutzerdefinierten Bereitstellungsrezepte ausgeführt werden.

> **Note**
>
> Weitere Informationen zur Fehlerbehebung bei fehlgeschlagenen Bereitstellungen oder Anwendungen finden Sie unter Debuggen von Rezepten.

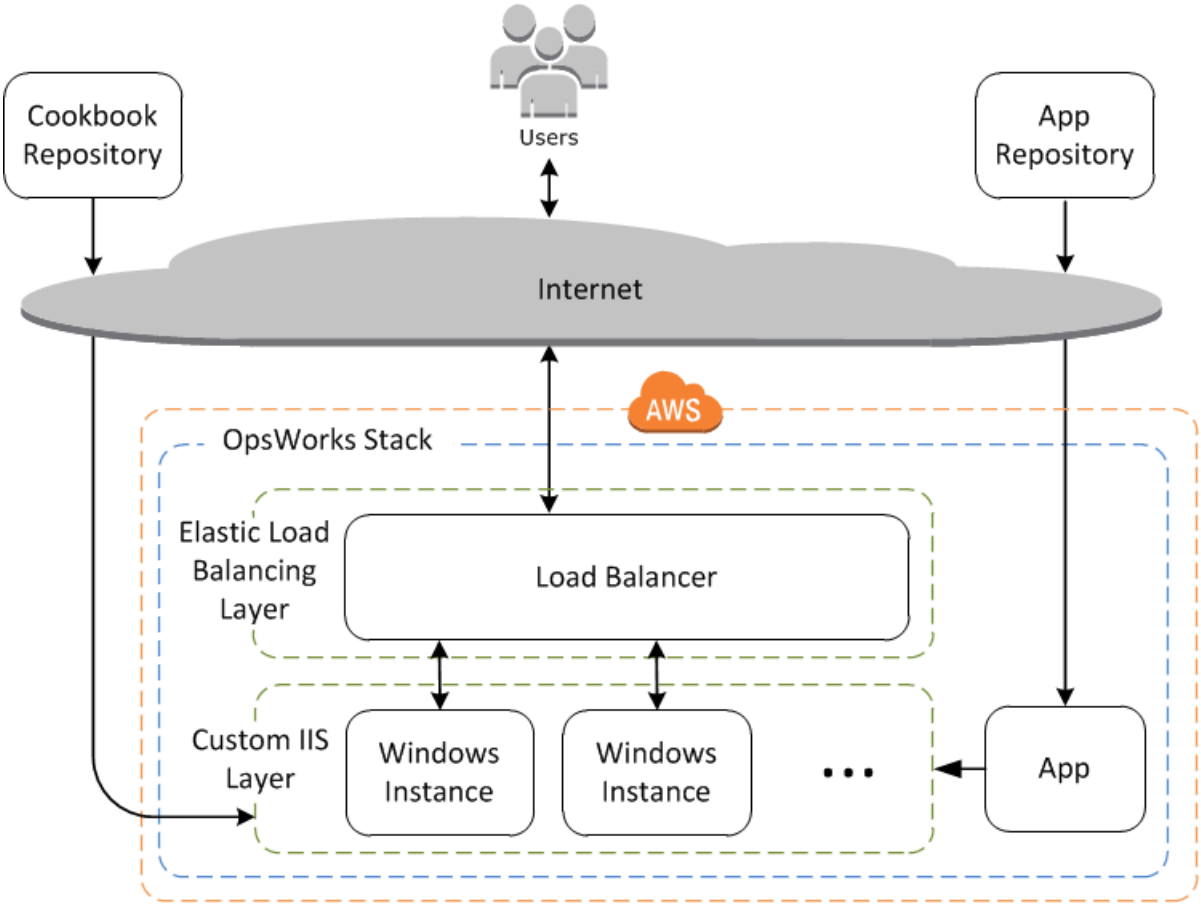
Die App ist jetzt installiert. Um sie zu öffnen, wählen Sie Instances (Instances) im Navigationsbereich und dann die öffentliche IP-Adresse der Instance aus. Es wird eine HTTP-Anfrage an die Instance gesendet und es wird in etwa Folgendes in Ihrem Browser angezeigt.

Hello World!

Schritt 3: Skalieren von IISExample

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn die eingehenden Benutzeranfragen eine einzelne t2.Micro-Instance nahezu auslasten, müssen Sie die Serverkapazitäten erweitern. Sie können auf eine größere Instance wechseln. Dieser Methode sind jedoch Grenzen gesetzt. Flexibler sind Sie, wenn Sie Ihrem Stack weitere Instances hinzufügen und einen Load Balancer davor schalten. Die grundlegende Architektur sieht dabei etwa wie folgt aus.



Dieser Ansatz hat unter anderem den Vorteil, dass er weniger fehleranfällig ist als eine einzelne große Instance.

- Falls eine Instance ausfällt, verteilt der Load Balancer eingehende Anfragen auf die übrigen Instances und die Anwendungen werden weiterhin ausgeführt.
- Wenn Sie Instances in verschiedenen Availability Zones betreiben (dies ist die empfohlene Vorgehensweise), funktionieren Anwendungen auch dann, wenn in einer Availability Zone Probleme auftreten.

AWS OpsWorks Stacks macht es einfach, Stapel zu skalieren. In diesem Abschnitt werden die Grundlagen beschrieben, wie Sie einen Stack skalieren können, indem Sie eine zweite rund um

die Uhr verfügbare PHP App Server-Instanz zu IISExample hinzufügen und beide Instanzen hinter einem Elastic Load Balancing Load Balancer platzieren. Sie können das Verfahren einfach erweitern, um eine beliebige Anzahl von 24/7-Instances hinzuzufügen, oder Sie können zeitbasierte Instances verwenden, um AWS OpsWorks Stacks Ihren Stack automatisch skalieren zu lassen. Weitere Informationen finden Sie unter Verwaltung der Last mit zeit- und lastbasierten Instanzen.

Hinzufügen eines Load Balancers

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Elastic Load Balancing ist ein AWS-Service, der den eingehenden Anwendungsdatenverkehr automatisch auf mehrere Amazon EC2 EC2-Instances verteilt. Ein Load Balancer hat zwei Anwendungsgebiete. Das offensichtlichere ist eine gleichmäßige Auslastung Ihrer Anwendungsserver zu gewährleisten. Viele Websites ziehen es vor, die Anwendungsserver und Datenbanken vom direkten Benutzerzugriff zu trennen. Elastic Load Balancing verteilt nicht nur den Traffic, sondern bietet auch folgende Funktionen:

- Erkennt fehlerhafte Amazon EC2 EC2-Instances.

  Er leitet Datenverkehr auf die übrigen fehlerfreien Instances um, bis die Fehler behoben wurden.
- Er skaliert als Reaktion auf den eingehenden Datenverkehr automatisch die Kapazität zur Anforderungsbearbeitung.

> **Note**
>
> AWS OpsWorks Stacks unterstützt den Application Load Balancer nicht. Sie können Classic Load Balancer nur mit AWS OpsWorks Stacks verwenden.

Elastic Load Balancing wird zwar oft als Ebene bezeichnet, funktioniert aber etwas anders als die anderen integrierten Ebenen. Anstatt eine Ebene zu erstellen und ihr Instances hinzuzufügen,

erstellen Sie mithilfe der Amazon EC2 EC2-Konsole einen Elastic Load Balancing Load Balancer und fügen ihn dann einer Ihrer vorhandenen Ebenen hinzu, normalerweise einer Anwendungsserverschicht. AWS OpsWorks Stacks registriert dann die vorhandenen Instances der Ebene beim Service und fügt automatisch alle neuen Instances hinzu. Nachfolgend wird beschrieben, wie Sie einen Load Balancer hinzufügen.

So fügen Sie einen Load Balancer an den benutzerdefinierten IIS-Layer an

1. Verwenden Sie die Amazon EC2 EC2-Konsole, um einen neuen Load Balancer für IISExample zu erstellen. Weitere Informationen finden Sie unter Erste Schritte mit Elastic Load Balancing. Wenn Sie den Assistenten Load Balancer erstellen ausführen, konfigurieren Sie den Load Balancer wie folgt:

   1: Definieren von Load Balancer

   Weisen Sie dem Load Balancer einen leicht erkennbaren Namen wie IIS-LB zu, damit er in der Stacks-Konsole leichter auffindbar ist. AWS OpsWorks Akzeptieren Sie die Standardwerte für die verbleibenden Einstellungen und wählen Sie dann Weiter: Zuweisen von Sicherheitsgruppen aus.

   2: Zuweisen von Sicherheitsgruppen

   Wenn Ihr Konto die Standard-VPC unterstützt, zeigt der Assistent diese Seite an, um die Sicherheitsgruppe des Load Balancers festzulegen. Für EC2 Classic wird diese Seite nicht angezeigt.

   Geben Sie für diese Anleitung default VPC security group (Standard-VPC-Sicherheitsgruppe) an und wählen Sie dann Weiter: Konfigurieren von Sicherheitseinstellungen aus.

   3: Konfigurieren von Sicherheitseinstellungen

   Für diese Anleitung müssen Sie für Ihren Load Balancer einen sicheren Listener verwenden (d. h. HTTPS oder SSL als Frontend-Verbindung). Wählen Sie daher Weiter: Konfigurieren der Zustandsprüfung aus, um fortzufahren.

   4: Konfigurieren der Zustandsprüfung

   Setzen Sie den Ping-Pfad auf **/**. Akzeptieren Sie die Standardwerte für die verbleibenden Einstellungen und wählen Sie dann Weiter: Hinzufügen von EC2-Instances aus.

5: Hinzufügen von EC2-Instances

AWS OpsWorks Stacks kümmert sich automatisch um die Registrierung von Instances beim Load Balancer. Wählen Sie Weiter: Hinzufügen von Tags aus, um fortzufahren.

6: Hinzufügen von Tags

In diesem Beispiel werden keine Tags verwendet. Wählen Sie Review and Create (Prüfen und Erstellen) aus.

7: Prüfen

Überprüfen Sie Ihre Auswahl und wählen Sie Create (Erstellen) und dann Close (Schließen) aus, um den Load Balancer zu starten.

2. Wenn Ihr Konto die Standard-VPC unterstützt, müssen Sie nach dem Starten des Load Balancers überprüfen, dass die Sicherheitsgruppe über die erforderlichen Regeln für eingehenden Datenverkehr verfügt. Mit der Standardregel wird eingehender Datenverkehr nicht akzeptiert.

   1. Wählen Sie im Amazon EC2 EC2-Navigationsbereich die Option Sicherheitsgruppen aus.
   2. Wählen Sie default VPC security group (Standard-VPC-Sicherheitsgruppe) aus.
   3. Klicken Sie auf die Registerkarte Inbound und wählen Sie Edit aus.
   4. Legen Sie für diese Anleitung für Source (Quelle) den Wert Anywhere (Beliebig) fest. So akzeptiert der Load Balancer eingehenden Datenverkehr von beliebigen IP-Adressen.
   5. Klicken Sie auf Speichern.
3. Kehren Sie zur AWS OpsWorks Stacks-Konsole zurück. Wählen Sie auf der Seite Layers (Ebenen) die Option Network (Netzwerk) aus.
4. Wählen Sie unter Elastic Load Balancing den IIS-LB-Load Balancer aus, den Sie in Schritt 1 erstellt haben, und klicken Sie auf Save (Speichern).

   Nachdem Sie den Load Balancer mit dem Layer verbunden haben, registriert AWS OpsWorks Stacks automatisch die aktuellen Instanzen des Layers und fügt neue Instanzen hinzu, sobald sie online sind.
5. Klicken Sie auf der Seite Layers (Ebenen) auf den Load Balancer-Namen, um dessen Detailseite aufzurufen. An dem grünen Häkchen neben der Instance auf der Seite des Load Balancers erkennen Sie, dass die Instance die Zustandsprüfung bestanden hat.

Sie können IIS-Example-App jetzt ausführen, indem Sie eine Anfrage an den Load Balancer senden.

So führen Sie IIS-Example-App über den Load Balancer aus

1. Wählen Sie Layers (Ebenen) aus. Der IIS-ELB-Load Balancer sollte als Layer aufgeführt sein und die Spalte "Zustandsprüfung" sollte eine grüne, also fehlerfreie Instance enthalten.
2. Wählen Sie den DNS-Namen des Load Balancers aus, um IIS-Example-App auszuführen. Die App sollte unter dem Namen des Load Balancers aufgeführt sein und etwa wie folgt aussehen: `IIS-LB-1802910859.us-west-2.elb.amazonaws.com`. Der Load Balancer leitet die Anfrage an die Instance weiter und gibt die Antwort zurück. Diese Antwort sollte mit derjenigen Antwort übereinstimmen, die Sie durch einen Klick auf die öffentliche IP-Adresse der Instance erhalten.

Sie haben bisher nur eine Instance, daher bringt der Load Balancer noch nicht viel. Sie können dem Layer jetzt jedoch weitere Instances hinzufügen.

So fügen Sie dem Layer eine Instance hinzu

1. Wählen Sie Instances und dann + instance (+ Instance) aus, um der Ebene eine weitere Instance hinzuzufügen.
2. Starten Sie die Instance.

Da es sich um neue Instanzen handelt, installiert AWS OpsWorks Stacks automatisch die aktuellen benutzerdefinierten Kochbücher und stellt während der Einrichtung die aktuelle App-Version bereit. Wenn die Instance online geht, fügt AWS OpsWorks Stacks sie automatisch dem Load Balancer hinzu, sodass Ihre Instance sofort mit der Bearbeitung von Anfragen beginnt. Um die Funktionsweise der Anwendung zu überprüfen, wählen Sie den DNS-Namen des Load Balancers erneut aus.

Nächste Schritte

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In dieser Anleitung haben Sie die Grundlagen der Einrichtung eines einfachen Windows-Anwendungsserver-Stacks kennengelernt. Im Folgenden finden Sie einige Vorschläge für die nächsten Schritte.

- Wenn du mehr wissen möchtest, Erste Schritte: Rezeptbücher bietet es ein Tutorial zur Einführung in die Implementierung von Kochbüchern und enthält eine Reihe von AWS OpsWorks Stacks-spezifischen Beispielen.
- Sie können dem Stack eine Amazon Relational Database Service (Amazon RDS) -Ebene hinzufügen, um sie als Backend-Datenbankserver zu verwenden. Weitere Informationen dazu, wie Sie Ihre Anwendung mit der Datenbank verknüpfen, finden Sie unter Verwenden von benutzerdefinierten Rezepten.

Erste Schritte mit Kochbüchern in Stapeln AWS OpsWorks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Ein AWS OpsWorks Stacks-Stack auf Produktionsebene erfordert in der Regel einige Anpassungen, was häufig die Implementierung eines benutzerdefinierten Chef-Kochbuches bedeutet. Ein Rezeptbuch ist eine Paket-Datei, die Konfigurationsdaten, einschließlich Anleitungen, die als Rezepte bezeichnet werden, enthält. Ein Rezept enthält mindestens eine in der Ruby-Sprachsyntax geschriebene Anleitung, die die zu verwendenden Ressourcen sowie deren Anwendungsreihenfolge angibt. Eine Ressource in Chef stellt eine Anweisung einer Konfigurationsrichtlinie dar. Diese exemplarische Vorgehensweise bietet eine grundlegende Einführung in die Implementierung von Chef-Kochbüchern für Stacks. AWS OpsWorks Weitere Informationen zu Chef, Rezeptbüchern, Rezepten und Ressourcen erhalten Sie über die folgenden Links: Nächste Schritte.

In dieser Anleitung wird im Wesentlichen das Erstellen eigener Rezeptbücher beschrieben. Sie können auch von der Community bereitgestellte Rezeptbücher auf Websites wie Chef Supermarket verwenden. Wir haben weiter hinten Anleitungen zur Nutzung eines Community-Rezeptbuchs

aus Chef Supermarket eingefügt, um Ihnen den Einstieg in die Rezeptbücher der Community zu erleichtern.

Ehe Sie mit dieser Anleitung beginnen, nehmen Sie noch einige Einrichtungsschritte vor. Wenn Sie bereits eine der anderen Anleitungen in diesem Kapitel durchgearbeitet haben, beispielsweise Erste Schritte: Beispiel, dann haben Sie die Voraussetzungen für diese Anleitung erfüllt und können direkt beginnen. Wenn Sie die Voraussetzungen noch nicht erfüllt haben, sollten Sie dies jetzt nachholen und anschließend zu dieser Anleitung zurückkehren.

Themen

Schritt 1: Erstellen des Rezeptbuchs

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Beginnen Sie, indem Sie ein Rezeptbuch erstellen. Dieses Rezeptbuch ist zunächst recht einfach gehalten, dient aber als Grundlage für den Rest dieser Anleitung.

> **Note**
>
> In diesem Schritt wird gezeigt, wie Sie ein Rezeptbuch manuell erstellen. Mit dem Chef Development Kit (Chef DK) können Sie Rezeptbücher schneller erstellen, indem Sie den Befehl chef generate cookbook auf Ihrer lokalen Workstation ausführen. Dieser Befehl erstellt allerdings mehrere Ordner und Dateien, die Sie für diese Anleitung nicht benötigen.

So erstellen Sie das Rezeptbuch

1. Erstellen Sie auf Ihrer lokalen Workstation ein Verzeichnis namens `opsworks_cookbook_demo`. Sie können grundsätzlich auch einen anderen Namen verwenden. Für diese Anleitung sollten Sie aber `opsworks_cookbook_demo` nutzen.
2. Erstellen Sie mithilfe eines Text-Editors im Verzeichnis `opsworks_cookbook_demo` eine Datei mit dem Namen `metadata.rb`. Fügen Sie den folgenden Code ein, um den Namen des Rezeptbuchs festzulegen. Weitere Informationen über `metadata.rb` finden Sie unter metadata.rb auf der Chef-Website.

   ```
   name "opsworks_cookbook_demo"
   ```

3. Erstellen Sie in dem Verzeichnis `opsworks_cookbook_demo` ein Unterverzeichnis namens `recipes`. In diesem Unterverzeichnis werden alle Rezepte gespeichert, die Sie für das Rezeptbuch dieser Anleitung erstellen.

4. Erstellen Sie in dem Verzeichnis `recipes` eine Datei namens `default.rb`. Diese Datei enthält ein Rezept mit demselben Namen wie die Datei, allerdings ohne die Dateierweiterung: `default`. Fügen Sie die folgende Codezeile zur Datei `default.rb` hinzu. Dieser Code ist ein Rezept, das nur aus einer Zeile besteht und bei Ausführung eine einfache Nachricht im Protokoll anzeigt:

   ```
   Chef::Log.info("********** Hello, World! **********")
   ```

5. Führen Sie am Terminal oder an der Eingabeaufforderung den Befehl tar aus, um eine Datei mit dem Namen `opsworks_cookbook_demo.tar.gz` zu erstellen, die das Verzeichnis `opsworks_cookbook_demo` und seinen Inhalt enthält. Beispielsweise:

   ```
   tar -czvf opsworks_cookbook_demo.tar.gz opsworks_cookbook_demo/
   ```

   Sie können grundsätzlich auch einen anderen Dateinamen verwenden. Für diese Anleitung sollten Sie aber `opsworks_cookbook_demo.tar.gz` nutzen.

   > **Note**
   >
   > Wenn Sie die `tar`-Datei unter Windows erstellt haben, muss das oberste Verzeichnis das übergeordnete Verzeichnis des Rezeptbuchs sein. Diese Anleitung wurde unter Linux mit dem Befehl tar getestet, der über das `tar`-Paket bereitgestellt wurde. Für Windows wurde der Befehl tar von Git Bash genutzt. Wenn Sie andere Befehle oder Programme zum Erstellen einer komprimierten TAR-Datei (.tar.gz) nutzen, erhalten Sie möglicherweise nicht das gewünschte Ergebnis.

6. Erstellen Sie einen S3-Bucket oder nutzen Sie einen vorhandenen Bucket. Weitere Informationen finden Sie unter Bucket erstellen.
7. Laden Sie die Datei `opsworks_cookbook_demo.tar.gz` in den S3-Bucket hoch. Weitere Informationen finden Sie unter Hinzufügen eines Objekts zu einem Bucket.

Sie verfügen jetzt über ein Rezeptbuch, mit dem Sie im weiteren Verlauf dieser Anleitung arbeiten werden.

Im nächsten Schritt erstellst du einen AWS OpsWorks Stacks-Stack, den du später verwenden wirst, um dein Kochbuch hochzuladen und die Rezepte des Kochbuches auszuführen.

Schritt 2: Erstellen des Stacks und dessen Komponenten

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Erstellen Sie einen AWS OpsWorks Stacks-Stack und seine Komponenten, zu denen eine Ebene und eine Instanz gehören. Später laden Sie zunächst Ihr Rezeptbuch in die Instance hoch und führen dann die Rezepte des Rezeptbuchs auf der Instance aus.

So erstellen Sie den Stack

1. Melden Sie sich bei der AWS OpsWorks Stacks-Konsole unter https://console.aws.amazon.com/opsworks an.
2. Führen Sie einen der folgenden Schritte aus, sofern erforderlich:
   - Wenn die Seite Willkommen bei AWS OpsWorks Stacks angezeigt wird, wählen Sie Add your first stack oder Add your first AWS OpsWorks Stacks stack (beide Optionen bewirken dasselbe). Die Seite Add stack (Stack hinzufügen) wird angezeigt.
   - Wenn die OpsWorks Dashboard-Seite angezeigt wird, wählen Sie Stapel hinzufügen. Die Seite Add Stack (Stack hinzufügen) wird angezeigt.
3. Wählen Sie Chef 12 stack aus.
4. Geben Sie im Feld Stack name den Stack-Namen ein, beispielsweise **`MyCookbooksDemoStack`**. Sie können grundsätzlich auch einen anderen Namen eingeben. Für diese Anleitung sollten Sie aber `MyCookbooksDemoStack` nutzen.
5. Wählen Sie für Region die Option US West (Oregon) aus.
6. Führen Sie für VPC einen der folgenden Schritte aus:
   - Wählen Sie eine VPC aus, falls verfügbar. Weitere Informationen finden Sie unter Ausführen eines Stacks in einer VPC.
   - Wählen Sie andernfalls No VPC (Keine VPC) aus.

7. Bestätigen Sie die Option Use custom Chef cookbooks (Benutzerdefinierte Chef-Rezeptbücher verwenden) mit Yes.
8. Wählen Sie für Repository type die Option S3 Archive aus.

   > **Note**
   >
   > In der Anleitung Erste Schritte: Linux haben Sie an dieser Stelle HTTP Archive ausgewählt. Wählen Sie hier stattdessen S3 Archive aus.

9. Geben Sie für Repository URL den Speicherort zur Datei `opsworks_cookbook_demo.tar.gz` in S3 an. Um den Pfad zu erhalten, wählen Sie in der S3-Konsole die Datei `opsworks_cookbook_demo.tar.gz` aus. Kopieren Sie im Bereich Properties (Eigenschaften) den Wert des Felds Link. (Es sollte dem Folgenden ähneln: `https://s3.amazonaws.com/opsworks-demo-bucket/opsworks_cookbook_demo.tar.gz`.)
10. Wenn Ihr S3-Bucket privat ist, was die Standardeinstellung ist, geben Sie für Access Key ID und Secret Access Key die Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel des IAM-Benutzers ein, den Sie für diese exemplarische Vorgehensweise verwenden. Weitere Informationen finden Sie unter Bearbeiten von Objektberechtigungen und Ein Objekt mit anderen teilen.
11. Übernehmen Sie die Standardeinstellungen für die folgenden Schritte:
    - Default Availability Zone (us-west-2a)
    - Standardbetriebssystem (Linux und Amazon Linux 2016.09)
    - Default SSH key (Do not use a default SSH key)
    - Stack color (dark blue)
12. Wählen Sie Advanced (Erweitert) aus.
13. Führen Sie für IAM role (IAM-Rolle) einen der folgenden Schritte aus:
    - Wählen Sie aws-opsworks-service-role aus, sofern verfügbar.
    - Falls aws-opsworks-service-rolenicht verfügbar, wählen Sie Neue IAM-Rolle.
14. Führen Sie für das Standard-IAM-Instanzprofil einen der folgenden Schritte aus:
    - Wenn aws-opsworks-ec2 Rollen verfügbar sind, wählen Sie sie aus.
    - Wenn aws-opsworks-ec2 Rollen nicht verfügbar sind, wählen Sie Neues IAM-Instanzprofil aus.
15. Übernehmen Sie die Standardeinstellungen für die folgenden Schritte:

- Default root device type (EBS backed)
- Hostname theme (Layer Dependent)
- OpsWorks Agentenversion (neueste Version)
- Custom Chef JSON (leer)
- Sicherheit, OpsWorks Sicherheitsgruppen verwenden (Ja)

16. Wählen Sie Stack hinzufügen. AWS OpsWorks Stacks erstellt den Stapel und zeigt die MyCookbooksDemoStackSeite an.

So erstellen Sie den Layer:

1. Wählen Sie im Service-Navigationsbereich Layers aus. Die Seite Layers wird angezeigt.
2. Wählen Sie Add a layer (Layer hinzufügen) aus.
3. Geben **`MyCookbooksDemoLayer`** Sie auf der OpsWorksRegisterkarte als Name den Text ein. Sie können grundsätzlich auch einen anderen Namen eingeben. Für diese Anleitung sollten Sie aber `MyCookbooksDemoLayer` nutzen.
4. Geben Sie für Short name **`cookbooks-demo`** ein. Sie können grundsätzlich auch einen anderen Namen eingeben. Für diese Anleitung sollten Sie aber `cookbooks-demo` nutzen.
5. Wählen Sie „Ebene hinzufügen“. AWS OpsWorks Stacks fügt die Ebene hinzu und zeigt die Seite „Ebenen“ an.

So erstellen und starten Sie eine Instance:

1. Wählen Sie im Service-Navigationsbereich Instances aus. Die Seite Instances wird angezeigt.
2. Wählen Sie Add an instance (Instance hinzufügen) aus.
3. Wählen Sie auf der Registerkarte New (Neu) die Option Advanced (Erweitert) aus.
4. Übernehmen Sie die Standardeinstellungen für die folgenden Schritte:

   - Hostname (cookbooks-demo1)
   - Size (c3.large)
   - Subnet (*`IP-Adresse`* us-west-2a)
   - Scaling type (24/7)
   - SSH key (Do not use a default SSH key)
   - Betriebssystem (Amazon Linux 2016.09)

- OpsWorks Agentenversion (vom Stack erben)
- Tenancy (Default - Rely on VPC settings)
- Root device type (EBS backed)
- Volume type (General Purpose (SSD))
- Volume size (8)

5. Wählen Sie Add instance (Instance hinzufügen) aus.
6. Für MyCookbooksDemoLayer, für cookbooks-demo1, für Actions wählen Sie Start. Fahren Sie nicht fort, bevor der Status zu online wechselt. Dieser Vorgang kann einige Minuten dauern. Haben Sie etwas Geduld.

Sie haben nun einen Stack, einen Layer und eine Instance, zu der das Rezeptbuch automatisch aus Ihrem S3-Bucket kopiert wurde. Im nächsten Schritt führen Sie das Standardrezept aus dem Rezeptbuch auf der Instance aus und testen es.

Schritt 3: Ausführen und Testen des Rezepts

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Führen Sie das `default` Rezept aus dem Kochbuch aus, das AWS OpsWorks Stacks in die Instanz kopiert hat, und testen Sie es. Wie Sie sich sicherlich erinnern, besteht dieses Rezept aus einer einzelnen Zeile und zeigt eine einfache Nachricht im Protokoll an, wenn es ausgeführt wird.

So führen Sie das Rezept aus

1. Wählen Sie im Service-Navigationsbereich Stack aus. Die Seite MyCookbooksDemoStack wird angezeigt.
2. Wählen Sie Run Command (Befehl ausführen) aus. Die Seite Run Command (Befehl ausführen) wird angezeigt.
3. Wählen Sie für Command (Befehl) die Option Execute Recipes (Rezepte ausführen) aus.

4. Geben Sie für Recipes to execute (Auszuführende Rezepte) **`opsworks_cookbook_demo::default`** ein.

   `opsworks_cookbook_demo` ist der Name des Rezeptbuchs, der in der Datei `metadata.rb` festgelegt ist. **`default`** ist der Name des auszuführenden Rezepts, also der Name der Datei `default.rb` im Unterverzeichnis `recipes` des Rezeptbuchs ohne Dateierweiterung.

5. Lassen Sie die folgenden Standardeinstellungen unverändert:

   - Comment (Kommentar) (leer)
   - Advanced, Custom Chef JSON (leer)
   - Instanzen (Alle ausgewählt, markiert, MyCookbooksDemoLayercookbooks-demo1 aktiviert)

6. Wählen Sie Execute Recipes (Rezepte ausführen) aus. Die Seite Running command execute_recipes (Befehl execute_recipes wird ausgeführt) wird angezeigt. Fahren Sie nicht fort, bevor der Status zu successful (erfolgreich) wechselt. Dieser Vorgang kann einige Minuten dauern. Haben Sie etwas Geduld.

So prüfen Sie die Rezeptergebnisse:

1. Wählen Sie bei geöffneter Seite Running command execute_recipes für cookbooks-demo1 und Log die Option show (anzeigen) aus. Die Protokollseite execute_recipes wird angezeigt.
2. Führen Sie im Protokoll einen Bildlauf nach unten durch und suchen Sie einen Eintrag, der dem folgenden ähnelt:

   ```
   [2015-11-13T19:14:39+00:00] INFO: ********** Hello, World! **********
   ```

Sie haben Ihr erstes Rezept erfolgreich ausgeführt! Im nächsten Schritt aktualisieren Sie Ihr Rezeptbuch durch Hinzufügen eines Rezepts, das ein Paket auf der Instance installiert.

Schritt 4: Aktualisieren des Rezeptbuchs zum Installieren eines Pakets

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Aktualisieren Sie Ihr Rezeptbuch, indem Sie ein Rezept hinzufügen, das auf der Instance ein Paket installiert, das den beliebten Texteditor GNU Emacs enthält.

Sie können sich zwar genauso einfach bei der Instanz anmelden und das Paket einmal installieren, aber wenn Sie ein Rezept schreiben, können Sie das Rezept einmal von AWS OpsWorks Stacks aus ausführen, um mehrere Pakete auf mehreren Instanzen in einem Stack gleichzeitig zu installieren.

So aktualisieren Sie das Rezeptbuch zum Installieren eines Pakets:

1. Erstellen Sie auf Ihrer lokalen Workstation im Unterverzeichnis `recipes` im Verzeichnis `opsworks_cookbook_demo` eine Datei namens `install_package.rb` mit dem folgenden Code:

   ```
   package "Install Emacs" do
     package_name "emacs"
   end
   ```

   Dieses Rezept installiert das `emacs`-Paket auf der Instance. (Weitere Informationen finden Sie unter package.)

   > **Note**
   >
   > Sie können dem Rezept einen beliebigen Dateinamen geben. Achten Sie nur darauf, den richtigen Rezeptnamen anzugeben, wann immer AWS OpsWorks Stacks das Rezept ausführen soll.

2. Führen Sie am Terminal oder an der Eingabeaufforderung den Befehl tar aus, um eine neue Version der Datei `opsworks_cookbook_demo.tar.gz` zu erstellen, die das Verzeichnis `opsworks_cookbook_demo` und seinen aktualisierten Inhalt enthält.
3. Laden Sie die aktualisierte Datei `opsworks_cookbook_demo.tar.gz` in Ihren S3-Bucket hoch.

Dieses neue Rezept wird ausgeführt, wenn Sie das Rezeptbuch auf der Instance aktualisieren und anschließend das neue Rezept aus dem aktualisierten Rezeptbuch ausführen. Der nächste Schritt beschreibt die notwendige Vorgehensweise.

Nachdem Sie den nächsten Schritt abgeschlossen haben, können Sie sich bei der Instance anmelden und an der Eingabeaufforderung emacs eingeben, um GNU Emacs zu starten. (Weitere Informationen finden Sie unter Verbinden Sie sich mit der Linux-Instanz.) Zum Beenden von GNU Emacs drücken Sie STRG+X und anschließend STRG+C.

> **Important**
>
> Um sich bei der Instance anzumelden, müssen Sie AWS OpsWorks Stacks zunächst Informationen über Ihren öffentlichen SSH-Schlüssel (den Sie mit Tools wie ssh-keygen oder PuTTYgen erstellen können) zur Verfügung stellen und anschließend Berechtigungen für den `MyCookbooksDemoStack` Stack festlegen, damit sich Ihr Benutzer bei der Instanz anmelden kann. Anweisungen finden Sie unter Registrierung des öffentlichen SSH-Schlüssels eines Benutzers und Anmelden mit SSH.

Schritt 5: Aktualisieren des Rezeptbuchs auf der Instance und Ausführen des Rezepts

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Aktualisieren Sie das Rezeptbuch auf der Instance und führen Sie das Rezept aus dem aktualisierten Rezeptbuch auf der Instance aus. Im weiteren Verlauf dieser Anleitung werden Sie diesen Schritt immer dann wiederholen, wenn Sie das Rezeptbuch durch Hinzufügen eines neuen Rezepts aktualisieren.

So aktualisieren Sie das Rezeptbuch auf der Instance:

1. Wählen Sie im Service-Navigationsbereich Stack aus. Die Seite MyCookbooksDemoStack wird angezeigt.

2. Wählen Sie Run Command (Befehl ausführen) aus. Die Seite Run Command (Befehl ausführen) wird angezeigt.
3. Wählen Sie für Command (Befehl) Update Custom Cookbooks (Benutzerdefinierte Rezeptbücher aktualisieren) aus.
4. Lassen Sie die folgenden Standardeinstellungen unverändert:
   - Comment (Kommentar) (leer)
   - Advanced, Custom Chef JSON (leer)
   - Erweitert, Instanzen (Alle auswählen aktiviert, aktiviert, MyCookbooksDemoLayercookbooks-demo1 aktiviert)
5. Wählen Sie Update Custom Cookbooks (Benutzerdefinierte Rezeptbücher aktualisieren) aus. Die Seite Running command update_custom_cookbooks (Befehl update_custom_cookbooks wird ausgeführt) wird angezeigt. Fahren Sie nicht fort, bevor der Status zu successful (erfolgreich) wechselt. Dieser Vorgang kann einige Minuten dauern. Haben Sie etwas Geduld.

So führen Sie das Rezept aus

1. Wählen Sie im Service-Navigationsbereich Stack aus. Die Seite MyCookbooksDemoStack wird angezeigt.
2. Wählen Sie Run Command (Befehl ausführen) aus. Die Seite Run Command (Befehl ausführen) wird angezeigt.
3. Wählen Sie für Command (Befehl) die Option Execute Recipes (Rezepte ausführen) aus.
4. Geben Sie für Recipes to execute den Namen des auszuführenden Rezepts an. Beim ersten Mal hat das Rezept die Bezeichnung **`opsworks_cookbook_demo::install_package`**.

   > **Note**
   >
   > Wenn Sie dieses Verfahren wiederholen, geben Sie den Namen des Rezeptbuchs (**`opsworks_cookbook_demo`**), gefolgt von zwei Doppelpunkten (**`::`**), gefolgt vom Namen des Rezepts (der Dateiname des Rezepts ohne die Dateierweiterung `.rb`) ein.

5. Lassen Sie die folgenden Standardeinstellungen unverändert:
   - Comment (Kommentar) (leer)
   - Advanced, Custom Chef JSON (leer)

- Instanzen Alle auswählen (aktiviert, markiert, cookbooks-demo1 MyCookbooksDemoLayeraktiviert)

6. Wählen Sie Execute Recipes (Rezepte ausführen) aus. Die Seite Running command execute_recipes (Befehl execute_recipes wird ausgeführt) wird angezeigt. Fahren Sie nicht fort, bevor der Status zu successful (erfolgreich) wechselt. Dieser Vorgang kann einige Minuten dauern. Haben Sie etwas Geduld.

> **Note**
>
> Sie müssen Rezepte nicht manuell ausführen. Sie können den Lebenszyklusereignissen einer Ebene Rezepte zuweisen, z. B. den Ereignissen Setup und Configure, und AWS OpsWorks Stacks führt diese Rezepte automatisch aus, wenn das Ereignis eintritt. Weitere Informationen finden Sie unter AWS OpsWorks Stapelt Lifecycle-Ereignisse.

Im nächsten Schritt aktualisieren Sie das Rezeptbuch, um einen Benutzer zur Instance hinzuzufügen.

Schritt 6: Aktualisieren des Rezeptbuchs zum Hinzufügen eines Benutzers

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Aktualisieren Sie Ihr Rezeptbuch, indem Sie ein Rezept hinzufügen, das einen lokalen Benutzer zur Instance hinzufügt und das Stammverzeichnis und die Shell für den Benutzer festlegt. Dies ähnelt der Ausführung der Linux-Befehle adduser bzw. useradd oder des Windows-Befehls net user. Sie fügen beispielsweise einen lokalen Benutzer zu einer Instance hinzu, wenn Sie den Zugriff auf die Dateien und Verzeichnisse der Instance steuern möchten.

Sie können Benutzer auch ohne Rezeptbücher verwalten. Weitere Informationen finden Sie unter Verwalten von Benutzern.

So aktualisieren Sie das Rezeptbuch auf der Instance und führen das neue Rezept aus:

1. Erstellen Sie auf Ihrer lokalen Workstation im Unterverzeichnis `recipes` im Verzeichnis `opsworks_cookbook_demo` eine Datei namens `add_user.rb` mit dem folgenden Code (weitere Informationen finden Sie unter user):

   ```
   user "Add a user" do
     home "/home/jdoe"
     shell "/bin/bash"
     username "jdoe"
   end
   ```

2. Führen Sie am Terminal oder an der Eingabeaufforderung den Befehl tar aus, um eine neue Version der Datei `opsworks_cookbook_demo.tar.gz` zu erstellen, die das Verzeichnis `opsworks_cookbook_demo` und seinen aktualisierten Inhalt enthält.
3. Laden Sie die aktualisierte Datei `opsworks_cookbook_demo.tar.gz` in Ihren S3-Bucket hoch.
4. Folgen Sie den Anweisungen in Schritt 5: Aktualisieren des Rezeptbuchs auf der Instance und Ausführen des Rezepts, um das Rezeptbuch auf der Instance zu aktualisieren und das Rezept auszuführen. Geben Sie im Schritt „Rezept ausführen” für Recipes to execute (Auszuführende Rezepte) **`opsworks_cookbook_demo::add_user`** ein.

So testen Sie das Rezept:

1. Melden Sie sich bei der Instance an, sofern Sie noch nicht angemeldet sind.
2. Führen Sie an der Eingabeaufforderung den folgenden Befehl aus, um das Hinzufügen des neuen Benutzers zu bestätigen:

   ```
   grep jdoe /etc/passwd
   ```

   Es werden Informationen ähnlich der folgenden zum Benutzer angezeigt, einschließlich Details wie Benutzername, ID-Nummer, Gruppen-ID-Nummer, Stammverzeichnis und Shell:

   ```
   jdoe:x:501:502::/home/jdoe:/bin/bash
   ```

Im nächsten Schritt aktualisieren Sie das Rezeptbuch, um ein Verzeichnis auf der Instance anzulegen.

Schritt 7: Aktualisieren des Rezeptbuchs, um ein Verzeichnis zu erstellen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Aktualisieren Sie Ihr Rezeptbuch, indem Sie ein Rezept hinzufügen, das ein Verzeichnis zur Instance hinzufügt. Dies ähnelt der Ausführung des Linux-Befehls mkdir oder der Windows-Befehle md bzw. mkdir .

So aktualisieren Sie das Rezeptbuch auf der Instance und führen das neue Rezept aus:

1. Erstellen Sie auf Ihrer lokalen Workstation im Unterverzeichnis `recipes` im Verzeichnis `opsworks_cookbook_demo` eine Datei namens `create_directory.rb` mit dem folgenden Code. Weitere Informationen finden Sie unter directory:

   ```
   directory "Create a directory" do
     group "root"
     mode "0755"
     owner "ec2-user"
     path "/tmp/create-directory-demo"
   end
   ```

2. Führen Sie am Terminal oder an der Eingabeaufforderung den Befehl tar aus, um eine neue Version der Datei `opsworks_cookbook_demo.tar.gz` zu erstellen, die das Verzeichnis `opsworks_cookbook_demo` und seinen aktualisierten Inhalt enthält.
3. Laden Sie die aktualisierte Datei `opsworks_cookbook_demo.tar.gz` in Ihren S3-Bucket hoch.
4. Folgen Sie den Anweisungen in Schritt 5: Aktualisieren des Rezeptbuchs auf der Instance und Ausführen des Rezepts, um das Rezeptbuch auf der Instance zu aktualisieren und das Rezept auszuführen. Geben Sie im Schritt „Rezept ausführen” für Recipes to execute (Auszuführende Rezepte) **`opsworks_cookbook_demo::create_directory`** ein.

So testen Sie das Rezept:

1. Melden Sie sich bei der Instance an, sofern Sie noch nicht angemeldet sind.
2. Führen Sie an der Eingabeaufforderung den folgenden Befehl aus, um das Hinzufügen des neuen Verzeichnisses zu bestätigen:

   ```
   ls -la /tmp/create-directory-demo
   ```

   Es werden Informationen zum neu hinzugefügten Verzeichnis angezeigt, einschließlich Daten zu Berechtigungen, Besitzername und Gruppenname:

   ```
   drwxr-xr-x 2 ec2-user root 4096 Nov 18 00:35 .
   drwxrwxrwt 6 root     root 4096 Nov 24 18:17 ..
   ```

Im nächsten Schritt aktualisieren Sie das Rezeptbuch, um eine Datei auf der Instance zu erstellen.

Schritt 8: Aktualisieren des Rezeptbuchs, um Dateien zu erstellen und zu kopieren

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Aktualisieren Sie Ihr Rezeptbuch, indem Sie ein Rezept hinzufügen, das zwei Dateien zur Instance hinzufügt. Die erste Ressource des Rezepts erstellt eine Datei vollständig mit Rezeptcode. Dies ähnelt der Ausführung der Linux-Befehle cat, echo bzw. touch oder der Windows-Befehle echo bzw. fsutil. Die Methode eignet sich, wenn Sie wenige, kleine oder einfache Dateien haben. Die zweite Ressource des Rezepts kopiert eine Datei aus dem Rezeptbuch in ein anderes Verzeichnis der Instance. Dies ähnelt der Ausführung des Linux-Befehls cp oder des Windows-Befehls copy. Diese Methode eignet sich, wenn Sie viele, große oder komplexe Dateien haben.

Schließen Sie Schritt 7: Aktualisieren des Rezeptbuchs, um ein Verzeichnis zu erstellen ab, um sicherzustellen, dass das übergeordnete Verzeichnis der Dateien bereits vorhanden ist, ehe Sie diesen Schritt durchführen.

So aktualisieren Sie das Rezeptbuch auf der Instance und führen das neue Rezept aus:

1. Erstellen Sie auf Ihrer lokalen Workstation im Verzeichnis `opsworks_cookbook_demo` ein Unterverzeichnis namens `files`.
2. Erstellen Sie im Unterverzeichnis `files` eine Datei mit dem Name `hello.txt` und dem folgenden Text: **`Hello, World!`**
3. Erstellen Sie im Unterverzeichnis `recipes` im Verzeichnis `opsworks_cookbook_demo` eine Datei namens `create_files.rb` mit dem folgenden Code. Weitere Informationen finden Sie unter file und 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
   ```

   Die `file`-Ressource erstellt eine Datei im angegebenen Pfad. Die `cookbook_file`-Ressource kopiert die Datei aus dem Verzeichnis `files`, das Sie gerade im Rezeptbuch erstellt haben, in ein anderes Verzeichnis auf der Instance (Chef erwartet ein Unterverzeichnis mit dem Standardnamen `files`, aus dem Dateien kopiert werden).
4. Führen Sie am Terminal oder an der Eingabeaufforderung den Befehl tar aus, um eine neue Version der Datei `opsworks_cookbook_demo.tar.gz` zu erstellen, die das Verzeichnis `opsworks_cookbook_demo` und seinen aktualisierten Inhalt enthält.
5. Laden Sie die aktualisierte Datei `opsworks_cookbook_demo.tar.gz` in Ihren S3-Bucket hoch.
6. Folgen Sie den Anweisungen in Schritt 5: Aktualisieren des Rezeptbuchs auf der Instance und Ausführen des Rezepts, um das Rezeptbuch auf der Instance zu aktualisieren und das Rezept

auszuführen. Geben Sie im Schritt „Rezept ausführen” für Recipes to execute (Auszuführende Rezepte) **`opsworks_cookbook_demo::create_files`** ein.

So testen Sie das Rezept:

1. Melden Sie sich bei der Instance an, sofern Sie noch nicht angemeldet sind.
2. Führen Sie an der Eingabeaufforderung nacheinander die folgenden Befehle aus, um das Hinzufügen der neuen Dateien zu bestätigen:

```
sudo cat /tmp/create-directory-demo/index.html

sudo cat /tmp/create-directory-demo/hello.txt
```

   Der Inhalt der Datei wird angezeigt:

```
<html>This is a placeholder for the home page.</html>

Hello, World!
```

Im nächsten Schritt aktualisieren Sie das Rezeptbuch, um einen Befehl auf der Instance auszuführen.

Schritt 9: Aktualisieren des Rezeptbuchs, um einen Befehl auszuführen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Aktualisieren Sie Ihr Rezeptbuch, indem Sie ein Rezept hinzufügen, das einen Befehl ausführt, der einen SSH-Schlüssel auf der Instance erstellt.

So aktualisieren Sie das Rezeptbuch auf der Instance und führen das neue Rezept aus:

1. Erstellen Sie auf Ihrer lokalen Workstation im Unterverzeichnis `recipes` im Verzeichnis `opsworks_cookbook_demo` eine Datei namens `run_command.rb` mit dem folgenden Code. Weitere Informationen finden Sie unter execute.

   ```
   execute "Create an SSH key" do
     command "ssh-keygen -f /tmp/my-key -N fLyC3jbY"
   end
   ```

2. Führen Sie am Terminal oder an der Eingabeaufforderung den Befehl tar aus, um eine neue Version der Datei `opsworks_cookbook_demo.tar.gz` zu erstellen, die das Verzeichnis `opsworks_cookbook_demo` und seinen aktualisierten Inhalt enthält.
3. Laden Sie die aktualisierte Datei `opsworks_cookbook_demo.tar.gz` in Ihren S3-Bucket hoch.
4. Folgen Sie den Anweisungen in Schritt 5: Aktualisieren des Rezeptbuchs auf der Instance und Ausführen des Rezepts, um das Rezeptbuch auf der Instance zu aktualisieren und das Rezept auszuführen. Geben Sie im Schritt „Rezept ausführen” für Recipes to execute (Auszuführende Rezepte) **`opsworks_cookbook_demo::run_command`** ein.

So testen Sie das Rezept:

1. Melden Sie sich bei der Instance an, sofern Sie noch nicht angemeldet sind.
2. Führen Sie an der Eingabeaufforderung nacheinander die folgenden Befehle aus, um das Erstellen des SSH-Schlüssels zu bestätigen:

   ```
   sudo cat /tmp/my-key

   sudo cat /tmp/my-key.pub
   ```

   Der Inhalt der privaten und öffentlichen SSH-Schlüssel wird angezeigt:

   ```
   -----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
```

Im nächsten Schritt aktualisieren Sie das Rezeptbuch, um ein Skript auf der Instance auszuführen.

Schritt 10: Aktualisieren des Rezeptbuchs, um ein Skript auszuführen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Aktualisieren Sie Ihr Rezeptbuch, indem Sie ein Rezept hinzufügen, das ein Skript auf der Instance ausführt. Dieses Rezept erstellt ein Verzeichnis und eine Datei in diesem Verzeichnis. Das Schreiben eines Rezepts zum Ausführen eines Skripts, das mehrere Befehle enthält, ist einfacher als das Ausführen der Befehle nacheinander.

So aktualisieren Sie das Rezeptbuch auf der Instance und führen das neue Rezept aus:

1. Erstellen Sie auf Ihrer lokalen Workstation im Unterverzeichnis `recipes` im Verzeichnis `opsworks_cookbook_demo` eine Datei namens `run_script.rb` mit dem folgenden Code. Weitere Informationen finden Sie unter 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. Führen Sie am Terminal oder an der Eingabeaufforderung den Befehl tar aus, um eine neue Version der Datei `opsworks_cookbook_demo.tar.gz` zu erstellen, die das Verzeichnis `opsworks_cookbook_demo` und seinen aktualisierten Inhalt enthält.

3. Laden Sie die aktualisierte Datei `opsworks_cookbook_demo.tar.gz` in Ihren S3-Bucket hoch.
4. Folgen Sie den Anweisungen in Schritt 5: Aktualisieren des Rezeptbuchs auf der Instance und Ausführen des Rezepts, um das Rezeptbuch auf der Instance zu aktualisieren und das Rezept auszuführen. Geben Sie im Schritt „Rezept ausführen" für Recipes to execute (Auszuführende Rezepte) **`opsworks_cookbook_demo::run_script`** ein.

So testen Sie das Rezept:

1. Melden Sie sich bei der Instance an, sofern Sie noch nicht angemeldet sind.
2. Führen Sie an der Eingabeaufforderung den folgenden Befehl aus, um das Hinzufügen der neuen Datei zu bestätigen:

   ```
   sudo cat /tmp/run-script-demo/helloworld.txt
   ```

   Der Inhalt der Datei wird angezeigt:

   ```
   Hello, World!
   ```

Im nächsten Schritt aktualisieren Sie das Rezeptbuch, um einen Service auf der Instance zu verwalten.

Schritt 11: Aktualisieren des Rezeptbuchs, um einen Service zu verwalten

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Aktualisieren Sie Ihr Rezeptbuch, indem Sie ein Rezept hinzufügen, das einen Service auf der Instance verwaltet. Dies ähnelt der Ausführung des Linux-Befehls service oder der Windows-Befehle net stop, net start und ähnlicher Befehle. Dieses Rezept stoppt den crond-Service auf der Instance.

So aktualisieren Sie das Rezeptbuch auf der Instance und führen das neue Rezept aus:

1. Erstellen Sie auf Ihrer lokalen Workstation im Unterverzeichnis `recipes` im Verzeichnis `opsworks_cookbook_demo` eine Datei namens `manage_service.rb` mit dem folgenden Code. Weitere Informationen finden Sie unter service.

   ```
   service "Manage a service" do
     action :stop
     service_name "crond"
   end
   ```

2. Führen Sie am Terminal oder an der Eingabeaufforderung den Befehl tar aus, um eine neue Version der Datei `opsworks_cookbook_demo.tar.gz` zu erstellen, die das Verzeichnis `opsworks_cookbook_demo` und seinen aktualisierten Inhalt enthält.
3. Laden Sie die aktualisierte Datei `opsworks_cookbook_demo.tar.gz` in Ihren S3-Bucket hoch.
4. Folgen Sie den Anweisungen in Schritt 5: Aktualisieren des Rezeptbuchs auf der Instance und Ausführen des Rezepts, um das Rezeptbuch auf der Instance zu aktualisieren und das Rezept auszuführen. Geben Sie im Schritt „Rezept ausführen” für Recipes to execute (Auszuführende Rezepte) **`opsworks_cookbook_demo::manage_service`** ein.

So testen Sie das Rezept:

1. Melden Sie sich bei der Instance an, sofern Sie noch nicht angemeldet sind.
2. Führen Sie an der Eingabeaufforderung den folgenden Befehl aus, um zu überprüfen, dass der crond-Service gestoppt wurde:

   ```
   service crond status
   ```

   Folgendes wird angezeigt:

   ```
   crond is stopped
   ```

3. Für einen Neustart des crond-Services führen Sie den folgenden Befehl aus:

   ```
   sudo service crond start
   ```

   Folgendes wird angezeigt:

```
Starting crond:  [  OK  ]
```

4. Zur Überprüfung, dass der crond-Service gestartet wurde, führen Sie den folgenden Befehl erneut aus:

   ```
   service crond status
   ```

   Es werden Informationen angezeigt, die den folgenden ähneln:

   ```
   crond (pid  3917) is running...
   ```

Im nächsten Schritt aktualisieren Sie das Rezeptbuch, um auf Informationen zu verweisen, die als ein benutzerdefiniertes JSON-Objekt auf der Instance gespeichert sind.

Schritt 12: Aktualisieren des Rezeptbuchs, um ein benutzerdefiniertes JSON-Objekt zu verwenden

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Aktualisieren Sie Ihr Rezeptbuch, indem Sie ein Rezept hinzufügen, das auf ein benutzerdefiniertes JSON-Objekt verweist, die auf der Instance gespeichert ist.

Sie können Informationen in einer benutzerdefinierten JSON angeben, wenn Sie einen Stack erstellen, aktualisieren oder klonen oder wenn Sie eine Bereitstellung oder einen Stack-Befehl ausführen. Dies ist hilfreich, wenn Sie beispielsweise eine kleinen, unveränderlichen Teil an Daten für Ihre Rezepte auf der Instance zur Verfügung stellen, anstatt diese Daten aus einer Datenbank abzurufen. Weitere Informationen finden Sie unter Nutzen eines benutzerdefinierten JSON-Objekts.

In dieser Anleitung nutzen Sie eine benutzerdefinierte JSON, um fiktive Informationen zu einer Kundenrechnung bereitzustellen. Das benutzerdefinierte JSON-Objekt wird später in diesem Schritt beschrieben.

So aktualisieren Sie das Rezeptbuch auf der Instance und führen das neue Rezept aus:

1. Erstellen Sie auf Ihrer lokalen Workstation im Unterverzeichnis `recipes` im Verzeichnis `opsworks_cookbook_demo` eine Datei namens `custom_json.rb`, die den folgenden Rezeptcode enthält:

   ```
   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']}' **********")
   ```

   Dieses Rezept zeigt Meldungen zu den Werten in der benutzerdefinierten JSON in einem Protokoll an.
2. Führen Sie am Terminal oder an der Eingabeaufforderung den Befehl tar aus, um eine neue Version der Datei `opsworks_cookbook_demo.tar.gz` zu erstellen, die das Verzeichnis `opsworks_cookbook_demo` und seinen aktualisierten Inhalt enthält.
3. Laden Sie die aktualisierte Datei `opsworks_cookbook_demo.tar.gz` in Ihren S3-Bucket hoch.
4. Folgen Sie den Anweisungen in Schritt 5: Aktualisieren des Rezeptbuchs auf der Instance und Ausführen des Rezepts, um das Rezeptbuch auf der Instance zu aktualisieren und das Rezept auszuführen. Geben Sie im Schritt „Rezept ausführen” für Recipes to execute (Auszuführende Rezepte) **`opsworks_cookbook_demo::custom_json`** ein. Geben Sie für Advanced (Erweitert), Custom Chef JSON (Benutzerdefiniertes Chef-JSON-Objekt), das folgende benutzerdefinierte JSON-Objekt ein:

   ```
   {
     "customer-id": "0123",
     "invoice-number": "9876",
     "line-items": {
       "line-1": "tractor",
       "line-2": "passenger car",
   ```

```
    "line-3": "trailer"
  }
}
```

So testen Sie das Rezept:

1. Wählen Sie , während die Seite Running command execute_recipes (Befehl execute_recipes wird ausgeführt) noch geöffnet ist, für cookbooks-demo1 und Log die Option show (anzeigen) aus. Die Protokollseite execute_recipes wird angezeigt.
2. Führen Sie im Protokoll einen Bildlauf nach unten durch, um die Einträge zu finden, die den folgenden ähneln:

   ```
   [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'
    **********
   ```

   Diese Einträge zeigen Informationen des benutzerdefinierten JSON-Objekts an, die im Feld Advanced, Custom Chef JSON eingegeben wurden.

Im nächsten Schritt aktualisierst du das Kochbuch, um Informationen aus Datenbeuteln zu erhalten. Dabei handelt es sich um Sammlungen von Stack-Einstellungen, die AWS OpsWorks Stacks für jede Instanz speichert.

Schritt 13: Aktualisieren des Rezeptbuchs, um Data Bags zu verwenden

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Aktualisiere dein Kochbuch, indem du ein Rezept hinzufügst, das auf die Stack-Einstellungen verweist, die AWS OpsWorks Stacks auf der Instanz in einer Reihe von Datenbeuteln speichert. Dieses Rezept zeigt im Protokoll Meldungen zu bestimmten auf der Instance gespeicherten Stack-Einstellungen an. Weitere Informationen hierzu finden Sie unter AWS OpsWorks Referenz für Stacks Data Bag.

So aktualisieren Sie das Rezeptbuch auf der Instance und führen das neue Rezept aus:

1. Erstellen Sie auf Ihrer lokalen Workstation im Unterverzeichnis `recipes` im Verzeichnis `opsworks_cookbook_demo` eine Datei namens `data_bags.rb`, die den folgenden Code enthält:

   ```
   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']}' **********")
   ```

   Dieses Rezept zeigt im Protokoll Meldungen zu bestimmten auf der Instance gespeicherten Stack-Einstellungen an.

2. Führen Sie am Terminal oder an der Eingabeaufforderung den Befehl tar aus, um eine neue Version der Datei `opsworks_cookbook_demo.tar.gz` zu erstellen, die das Verzeichnis `opsworks_cookbook_demo` und seinen aktualisierten Inhalt enthält.
3. Laden Sie die aktualisierte Datei `opsworks_cookbook_demo.tar.gz` in Ihren S3-Bucket hoch.
4. Folgen Sie den Anweisungen in Schritt 5: Aktualisieren des Rezeptbuchs auf der Instance und Ausführen des Rezepts, um das Rezeptbuch auf der Instance zu aktualisieren und das Rezept auszuführen. Geben Sie im Schritt „Rezept ausführen” für Recipes to execute (Auszuführende Rezepte) **`opsworks_cookbook_demo::data_bags`** ein.

So testen Sie das Rezept:

1. Wählen Sie, während die Seite Running command execute_recipes (Befehl execute_recipes wird ausgeführt) noch geöffnet ist, für cookbooks-demo1 und Log die Option show (anzeigen) aus. Die Protokollseite execute_recipes wird angezeigt.
2. Führen Sie im Protokoll einen Bildlauf nach unten durch, um die Einträge zu finden, die den folgenden ähneln:

   ```
   [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/opsworks-demo-bucket/opsworks_cookbook_demo.tar.gz'
    **********
   ```

   Dieses Rezept zeigt Meldungen zu bestimmten auf der Instance gespeicherten Stack-Einstellungen an.

Im nächsten Schritt aktualisieren Sie das Rezeptbuch, um einen Rezeptcode mehrmals auszuführen.

Schritt 14: Aktualisieren des Rezeptbuchs, um Iterationsmethoden zu verwenden

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Aktualisieren Sie Ihr Rezeptbuch, indem Sie ein Rezept hinzufügen, das eine Iteration einsetzt. Bei dieser Methode wird der Rezeptcode mehrmals wiederholt. Das Rezept zeigt im Protokoll Meldungen zu einem Data Bag-Element an, das verschiedene Inhalte aufweist.

So aktualisieren Sie das Rezeptbuch auf der Instance und führen das neue Rezept aus:

1. Erstellen Sie auf Ihrer lokalen Workstation im Unterverzeichnis `recipes` im Verzeichnis `opsworks_cookbook_demo` eine Datei namens `iteration_demo.rb`, die den folgenden Code enthält:

   ```
   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**
   >
   > Das Schreiben des vorherigen Rezepts geht schneller, ist flexibler und weniger fehleranfällig als das Schreiben des folgenden Rezeptcodes, bei dem keine Iterationsmethoden verwendet werden:
   >
   > ```
   > 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. Führen Sie am Terminal oder an der Eingabeaufforderung den Befehl tar aus, um eine neue Version der Datei `opsworks_cookbook_demo.tar.gz` zu erstellen, die das Verzeichnis `opsworks_cookbook_demo` und seinen aktualisierten Inhalt enthält.
3. Laden Sie die aktualisierte Datei `opsworks_cookbook_demo.tar.gz` in Ihren S3-Bucket hoch.

4. Folgen Sie den Anweisungen in Schritt 5: Aktualisieren des Rezeptbuchs auf der Instance und Ausführen des Rezepts, um das Rezeptbuch auf der Instance zu aktualisieren und das Rezept auszuführen. Geben Sie im Schritt „Rezept ausführen" für Recipes to execute (Auszuführende Rezepte) **`opsworks_cookbook_demo::iteration_demo`** ein.

So testen Sie das Rezept:

1. Wählen Sie , während die Seite Running command execute_recipes (Befehl execute_recipes wird ausgeführt) noch geöffnet ist, für cookbooks-demo1 und Log die Option show (anzeigen) aus. Die Protokollseite execute_recipes wird angezeigt.
2. Führen Sie im Protokoll einen Bildlauf nach unten durch, um die Einträge zu finden, die den folgenden ähneln:

```
[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/
opsworks-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]' **********
```

   Das Rezept zeigt im Protokoll Meldungen zu einem Data Bag-Element an, das verschiedene Inhalte aufweist. Das Data Bag-Element befindet sich im `aws_opsworks_stack`-Data Bag. Das Data Bag-Element hat einen Inhalt namens `custom_cookbooks_source`. Der Inhalt ist in sechs Inhalte mit den Bezeichnungen `type`, `urlusername`, `password`, `ssh_key` und `revision` unterteilt, dessen Werte ebenfalls angezeigt werden.

Im nächsten Schritt aktualisieren Sie das Rezeptbuch, so dass ein Rezeptcode nur ausgeführt wird, wenn bestimmte Bedingungen erfüllt sind.

Schritt 15: Aktualisieren des Rezeptbuchs, um eine Bedingungslogik zu verwenden

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Aktualisieren Sie jetzt Ihr Rezeptbuch, indem Sie ein Rezept hinzufügen, das eine Bedingungslogik verwendet. Bei dieser Methode wird der Code nur ausgeführt, wenn bestimmte Bedingungen erfüllt sind. Weitere Informationen finden Sie unter if Statements und case Statements.

Je nach Data Bag-Inhalt ermöglicht dieses Rezept Folgendes: Es zeigt eine Meldung im Protokoll zur Identifizierung des Betriebssystems an, auf dem die Instance ausgeführt wird. Zudem installiert es ein Paket mit dem richtigen Paket-Manager für die vorhandene Linux-Verteilung. Dieses Paket erhält den Namen tree. Es ist eine einfache Anwendung zum Visualisieren von Verzeichnislisten.

So aktualisieren Sie das Rezeptbuch auf der Instance und führen das neue Rezept aus:

1. Erstellen Sie auf Ihrer lokalen Workstation im Unterverzeichnis `recipes` im Verzeichnis `opsworks_cookbook_demo directory` eine Datei namens `conditional_logic.rb`, die den folgenden Code enthält:

```
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. Führen Sie am Terminal oder an der Eingabeaufforderung den Befehl tar aus, um eine neue Version der Datei `opsworks_cookbook_demo.tar.gz` zu erstellen, die das Verzeichnis `opsworks_cookbook_demo` und seinen aktualisierten Inhalt enthält.
3. Laden Sie die aktualisierte Datei `opsworks_cookbook_demo.tar.gz` in Ihren S3-Bucket hoch.
4. Folgen Sie den Anweisungen in Schritt 5: Aktualisieren des Rezeptbuchs auf der Instance und Ausführen des Rezepts, um das Rezeptbuch auf der Instance zu aktualisieren und das Rezept auszuführen. Geben Sie im Schritt „Rezept ausführen” für Recipes to execute (Auszuführende Rezepte) **`opsworks_cookbook_demo::conditional_logic`** ein.

So testen Sie das Rezept:

1. Wählen Sie , während die Seite Running command execute_recipes (Befehl execute_recipes wird ausgeführt) noch geöffnet ist, für cookbooks-demo1 und Log die Option show (anzeigen) aus. Die Protokollseite execute_recipes wird angezeigt.
2. Führen Sie im Protokoll einen Bildlauf nach unten durch und suchen Sie einen Eintrag, der dem folgenden ähnelt:

```
[2015-11-16T19:59:05+00:00] INFO: ********** Operating system is Amazon Linux.
 **********
```

Da das Betriebssystem der Instance Amazon Linux 2016.09 ist, wird nur der vorherige Eintrag (von den fünf möglichen Einträgen im Code des Rezepts) im Protokoll angezeigt.

3. Wenn das Betriebssystem Linux ist, erstellt das Rezept das Paket tree. Zum Anzeigen der Verzeichnisinhalte geben Sie an der Eingabeaufforderung des gewünschten Verzeichnisses **`tree`** ein oder nutzen den gewünschten Verzeichnispfad (beispielsweise `tree /var/chef/runs`).

Im nächsten Schritt aktualisieren Sie das Rezeptbuch, um Funktionen eines externen Rezeptbuchs zu nutzen, das von der Chef-Community bereitgestellt wird.

Schritt 16: Aktualisieren des Rezeptbuchs, um Community-Rezeptbücher zu verwenden

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Aktualisieren Sie abschließend das Rezeptbuch, um Funktionen aus einem externen Rezeptbuch zu verwenden, das von der Chef-Community bereitgestellt wird. Das für diese Anleitung notwendige externe Rezeptbuch erhalten Sie im Chef Supermarket, einer beliebten Quelle für externe Chef-Rezeptbücher. Dieses externe Rezeptbuch stellt eine benutzerdefinierte Ressource zur Verfügung, über die Sie Anwendungen herunterladen und installieren können, ähnlich dem, was Sie in Schritt 4: Aktualisieren des Rezeptbuchs zum Installieren eines Pakets durchgeführt haben. Über diese Ressource können neben Paketen auch Webanwendungen und andere Anwendungstypen installiert werden.

Wenn ein Rezeptbuch von einem anderen Rezeptbuch abhängt, müssen Sie für das andere Rezeptbuch eine Abhängigkeit angeben. Wir empfehlen für das Deklarieren und Verwalten von

Rezeptbuch-Abhängigkeiten ein Tool namens Berkshelf. Weitere Informationen zum Installieren von Berkshelf auf Ihrer lokalen Workstation finden Sie unter About Berkshelf auf der Chef-Website.

Folgen Sie nach der Installation von Berkshelf diesen Anweisungen, um die Rezeptbuch-Abhängigkeit zu deklarieren. Erstellen Sie dann ein Rezept, das die Ressource in einem externen Rezeptbuch aufruft:

So deklarieren Sie die Rezeptbuch-Abhängigkeit:

1. Fügen Sie auf Ihrer lokalen Workstation im Verzeichnis `opsworks_cookbook_demo` folgende Zeile am Ende der Datei `metadata.rb` hinzu:

   ```
   depends "application", "5.0.0"
   ```

   Dies deklariert eine Abhängigkeit von einem Rezeptbuch namens `application`, Version 5.0.0.

2. Führen Sie vom Stamm des Verzeichnisses `opsworks_cookbook_demo` folgenden Befehl aus: Der Punkt am Ende des Befehls ist beabsichtigt.

   ```
   berks init .
   ```

   Berkshelf erstellt eine Reihe von Ordnern und Dateien, die Sie später für erweiterte Szenarien verwenden können. Die einzige Datei, die wir für diese Anleitung benötigen, ist die Datei `Berksfile`.

3. Fügen Sie die folgende Zeile am Ende der `Berksfile`-Datei hinzu:

   ```
   cookbook "application", "5.0.0"
   ```

   Dies informiert Berkshelf darüber, dass Sie das Anwendungs-Rezeptbuch, Version 5.0.0, verwenden möchten, auf Berkshelf von Chef Supermarket herunterlädt.

4. Führen Sie am Terminal oder der Eingabeaufforderung den folgenden Befehl vom Stamm der Verzeichnisses `opsworks_cookbook_demo` aus:

   ```
   berks install
   ```

   Berkshelf erstellt eine Liste von Abhängigkeiten für Ihr Rezeptbuch und das der Anwendung. Berkshelf benötigt diese Liste der Abhängigkeiten für den folgenden Vorgang.

So aktualisieren Sie das Rezeptbuch auf der Instance und führen das neue Rezept aus:

1. Erstellen Sie im Unterverzeichnis `recipes` im Verzeichnis `opsworks_cookbook_demo` eine Datei namens `dependencies_demo.rb`, das folgenden Code enthält:

   ```
   application "Install NetHack" do
     package "nethack.x86_64"
   end
   ```

   Dieses Rezept hängt von der Anwendungsressource aus dem Anwendungskochbuch ab, um das beliebte textbasierte Abenteuerspiel NetHack auf der Instanz zu installieren. (Sie können auch einen anderen Paketnamen verwenden, sofern das Paket für den Paket-Manager auf der Instance zur Verfügung steht.)

2. Führen Sie vom Stamm des Verzeichnisses `opsworks_cookbook_demo` folgenden Befehl aus:

   ```
   berks package
   ```

   Berkshelf verwendet die Liste der Abhängigkeiten aus dem vorherigen Verfahren, um eine Datei namens `cookbooks-`*`timestamp`*`.tar.gz` zu erstellen, die das Verzeichnis `opsworks_cookbook_demo` und dessen aktualisierte Inhalte, einschließlich der vom Rezeptbuch abhängigen Rezeptbücher, enthält. Benennen Sie diese Datei in `opsworks_cookbook_demo.tar.gz` um.

3. Laden Sie die aktualisierte, umbenannte Datei `opsworks_cookbook_demo.tar.gz` in Ihren S3-Bucket hoch.

4. Folgen Sie den Anweisungen in Schritt 5: Aktualisieren des Rezeptbuchs auf der Instance und Ausführen des Rezepts, um das Rezeptbuch auf der Instance zu aktualisieren und das Rezept auszuführen. Geben Sie im Schritt „Rezept ausführen” für Recipes to execute (Auszuführende Rezepte) **`opsworks_cookbook_demo::dependencies_demo`** ein.

5. Nach Ausführung des Rezepts sollten Sie sich bei der Instance anmelden und an der Eingabeaufforderung **`nethack`** eingeben können, um das Spiel zu starten. (Weitere Informationen zum Spiel finden Sie unter NetHackund im NetHackLeitfaden.)

Im nächsten Schritt können Sie die AWS Ressourcen bereinigen, die Sie für diese Komplettlösung verwendet haben. Dieser nächste Schritt ist optional. Möglicherweise möchten Sie diese AWS Ressourcen weiterhin verwenden, wenn Sie mehr über AWS OpsWorks Stacks erfahren. Wenn Sie diese AWS Ressourcen behalten, kann dies jedoch zu laufenden Gebühren für Ihr AWS Konto

führen. Wenn Sie diese AWS Ressourcen für eine spätere Verwendung behalten möchten, haben Sie diese exemplarische Vorgehensweise nun abgeschlossen, und Sie können direkt Nächste Schritte weitermachen.

Schritt 17: (Optional) Bereinigen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Um zu verhindern, dass zusätzliche Gebühren für Ihr AWS Konto anfallen, können Sie die AWS Ressourcen löschen, die für diese Komplettlösung verwendet wurden. Zu diesen AWS Ressourcen gehören der S3-Bucket, der AWS OpsWorks Stacks-Stack und die Komponenten des Stacks. (Weitere Informationen finden Sie unter OpsWorks AWS-Preise.) Möglicherweise möchten Sie diese AWS Ressourcen jedoch weiterhin nutzen, um mehr über AWS OpsWorks Stacks zu erfahren. Wenn Sie diese AWS Ressourcen weiterhin verfügbar halten möchten, haben Sie diese exemplarische Vorgehensweise nun abgeschlossen, und Sie können weitermachen. Nächste Schritte

Inhalte, die in den Ressourcen gespeichert sind, die Sie für diese schrittweise Anleitung erstellt haben, können persönlich identifizierende Informationen enthalten. Wenn Sie nicht mehr möchten, dass diese Informationen von AWS gespeichert werden, führen Sie die in diesem Thema beschriebenen Schritte aus.

So löschen Sie den S3-Bucket:

- Siehe Löschen des Amazon S3-Buckets.

So löschen Sie die Instance für den Stack

1. Wählen Sie in der AWS OpsWorks Stacks-Konsole im Service-Navigationsbereich Instances aus. Die Seite Instances wird angezeigt.
2. Wählen Sie für MyCookbooksDemoLayercookbooks-demo1 für Actions die Option stop aus. Wählen Sie im Bestätigungsdialogfeld Stop aus.

3. Die folgenden Änderungen können einige Minuten dauern. Fahren Sie erst fort, wenn alle im Folgenden genannten Änderungen abgeschlossen sind.

   - Der Status ändert sich von online zu stopping (wird angehalten) und schließlich zu stopped (angehalten).
   - online ändert sich von 1 zu 0.
   - shutting down ändert sich von 0 zu 1 und schließlich wieder zu 0.
   - stopped ändert sich schließlich von 0 zu 1.
4. Wählen Sie bei Actions (Aktionen) die Option delete (löschen) aus. Wenn Sie die Bestätigungsnachricht sehen, wählen Sie Löschen. AWS OpsWorks Stacks löscht die Instanz und zeigt Keine Instanzen an.

So löschen Sie den Stack

1. Wählen Sie im Service-Navigationsbereich Stack aus. Die Seite MyCookbooksDemoStack wird angezeigt.
2. Wählen Sie Delete Stack. Wenn Sie die Bestätigungsmeldung sehen, wählen Sie Löschen. AWS OpsWorks Stacks löscht den Stapel und zeigt die Dashboard-Seite an.

Optional können Sie das IAM-Benutzer- und Amazon EC2 EC2-Schlüsselpaar, das Sie für diese exemplarische Vorgehensweise verwendet haben, löschen, wenn Sie sie nicht für den Zugriff auf andere AWS Services und EC2-Instances wiederverwenden möchten. Anweisungen finden Sie unter Löschen eines IAM-Benutzers und Amazon EC2 EC2-Schlüsselpaare und Linux-Instances.

Sie haben diese Anleitung nun abgeschlossen. Weitere Informationen finden Sie unter Nächste Schritte.

Nächste Schritte

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Nachdem Sie diese exemplarische Vorgehensweise abgeschlossen haben, können Sie in den folgenden Ressourcen mehr über die AWS OpsWorks Stacks-Unterstützung für Chef-Kochbücher erfahren:

- Cookbooks und Rezepte— Beschreibt die Versionen von Chef und Ruby, die AWS OpsWorks Stacks derzeit unterstützt. Veranschaulicht zudem das Installieren und Aktualisieren von benutzerdefinierten Rezeptbüchern auf Instances sowie das Ausführen von Rezepten auf diesen.
- Learn Chef – Bietet Links zu Chef-Tutorials, einer Bibliothek mit Chef-Funktionen, einer vollständigen Chef-Dokumentation und Chef-Schulungen.
- Alles über Chef — Bietet eine vollständige Chef-Dokumentation. Spezifische Themen von Interesse sind:
  - About Cookbooks – Beschreibt die wichtigsten Rezeptbuch-Komponenten wie Attribute, Rezepte, Dateien, Metadaten und Vorlagen.
  - About Recipes – Beschreibt die Grundlagen von Rezepten, beispielsweise die Arbeit mit Data Bags, das Einschließen weiterer Rezepte und das Verwenden von Ruby-Code in Rezepten.
  - Resources – Beschreibt, wie alle integrierten Chef-Ressourcen wie `apt_package`, `cookbook_filedirectory`, `execute`, `file` und `package` verwendet werden.
  - About the Recipe DSL – Beschreibt das Schreiben von Code für Chef-Rezepte mit Anweisungen wie `if`, `case`, `data_bag`, `data_bag_item` und `search`.
- About Templates – Beschreibt, wie eingebettete Ruby (ERB)-Vorlagen verwendet werden, um dynamisch statische Textdateien wie Konfigurationsdateien zu generieren.
- Learning Tracks — Beschreibt, wie Sie Chef verwenden, um eine Instanz zu verwalten, eine grundlegende Web-App zu verwalten, Infrastrukturcode zu entwickeln und zu testen, Chef Analytics zu verwenden und vieles mehr.
- Learning Chef — Eine Einführung in Chef. Veröffentlicht von O'Reilly Media.
- Learning Chef code examples – Bietet begleitende Code-Beispiele für das von O'Reilly Media herausgegebene Buch Learning Chef.

AWS OpsWorks Bewährte Methoden für Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir

empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Strategien, Techniken und Vorschläge in diesem Abschnitt können Ihnen helfen, den größtmöglichen Nutzen und optimale Ergebnisse aus AWS OpsWorks Stacks zu ziehen.

Themen

Bewährte Methoden: Root-Gerätespeicher für Instances

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Dieses Thema gilt nicht für Windows-Instances, die von Amazon Elastic Block Store unterstützt werden müssen.

Amazon Elastic Compute Cloud (Amazon EC2) Linux-Instances haben die folgenden Root-Device-Speicheroptionen.

- Instance-Store-Backed Instances — Das Root-Gerät ist temporär.

Wenn Sie die Instance anhalten, gehen die Daten auf dem Root-Gerät verloren und können nicht wiederhergestellt werden. Weitere Informationen finden Sie unter Amazon EC2-Instance-Speicher.

- Amazon EBS-gestützte Instances — Das Root-Gerät ist ein Amazon EBS-Volume.

  Wenn Sie die Instance beenden, bleibt das Amazon EBS-Volume bestehen. Wenn Sie die Instance neu starten, wird das Volume automatisch neu aufgespielt und stellt den Instance-Status und alle gespeicherten Daten wieder her. Sie können das Volume auch auf eine andere Instance aufspielen. Weitere Informationen finden Sie unter Amazon Elastic Block Store (Amazon EBS).

Beachten Sie Folgendes bei der Wahl der zu nutzenden Root-Gerätespeicheroption.

Startzeit

Nach dem ersten Start werden Amazon EBS-Instances in der Regel schneller neu gestartet.

Der erste Start dauert bei beiden Speichertypen in etwa gleich lang. Beide Typen müssen eine vollständige Einrichtung durchführen, was relativ zeitaufwendige Aufgaben wie das Installieren von Paketen von Remote-Repositorys einschließt. Beachten Sie jedoch diese Unterschiede, wenn Sie eine Instance später neu starten:

- Durch Instance-Speicher gesicherte Instances führen dieselben Einrichtungsaufgaben durch wie beim ersten Start, einschließlich Paketinstallation.

  Ein Neustart dauert etwa gleich lang wie der erste Start.

- Amazon EBS-Back-Instances stellen das Root-Volume erneut bereit und führen die Setup-Rezepte aus.

  Der Neustart ist in der Regel wesentlich schneller als der Erststart, da die Einrichtungsrezepte Aufgaben wie das Neuinstallieren von Paketen, die bereits auf dem Stamm-Volume installiert sind, nicht durchführen müssen.

Kosten

Amazon EBS-gestützte Instances sind teurer:

- Mit einer vom Instance-Speicher gestützten Instance zahlen Sie nur, während die Instance ausgeführt wird.
- Bei Amazon EBS-gestützten Instances zahlen Sie für das Amazon EBS-Volume, unabhängig davon, ob die Instance läuft oder nicht.

Weitere Informationen finden Sie in der Preisübersicht zu Amazon EBS.

Protokollierung

Amazon EBS-gestützte Instances speichern automatisch Protokolle:

- Mit der Instance-Speicher gestützten Instance verschwinden die Protokolle, wenn die Instance stoppt.

  Sie müssen entweder die Protokolle abrufen, bevor Sie die Instance beenden, oder einen Dienst wie CloudWatch Logs verwenden, um ausgewählte Protokolle remote zu speichern.
- Bei einer Amazon EBS-gestützten Instance werden die Protokolle auf dem Amazon EBS-Volume gespeichert.

  Sie werden durch einen Neustart der Instance oder durch das Aufspielen des Volume auf eine anderen Instanz angezeigt.

Abhängigkeiten

Die beiden Speichertypen verfügen über verschiedene Abhängigkeiten:

- Instances, die von Instance-Stores unterstützt werden, hängen von Amazon S3 ab.

  Wenn Sie die Instance starten, muss sie das AMI von Amazon S3 herunterladen.
- Amazon EBS-gestützte Instances hängen von Amazon EBS ab.

  Wenn Sie die Instance starten, muss sie das Amazon EBS-Root-Volume mounten.

Empfehlung: Wenn Sie sich nicht sicher sind, welcher Speichertyp für Ihre Anforderungen am besten geeignet ist, empfehlen wir, mit Amazon EBS-Instances zu beginnen. Obwohl Ihnen für die Amazon EBS-Volumes geringe Kosten entstehen, ist das Risiko eines unbeabsichtigten Datenverlusts geringer.

Bewährte Methoden: Optimieren der Anzahl der Anwendungsserver

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Ein Produktions-Stack enthält mehrere Anwendungsserver, verteilt über mehrere Availability Zones. Allerdings kann die Anzahl der eingehenden Anforderungen je nach Tageszeit oder Wochentag erheblich variieren. Sie können einfach genügend Server betreiben, um die maximale erwartete Last zu verarbeiten, aber dann werden Sie die meiste Zeit für eine größere Serverkapazität bezahlen, als Sie benötigen. Um Ihre Website effizient zu betreiben, wird empfohlen, dass die Anzahl der Server dem aktuellen Anforderungsvolumen entspricht.

AWS OpsWorks Stacks bietet drei Möglichkeiten, die Anzahl der Serverinstanzen zu verwalten.

- 24/7-Instances werden manuell gestartet und laufen, bis sie manuell gestoppt werden.
- Zeitbasierte Instanzen werden von AWS OpsWorks Stacks automatisch nach einem vom Benutzer festgelegten Zeitplan gestartet und gestoppt.
- Lastbasierte Instances werden von AWS OpsWorks Stacks automatisch gestartet und gestoppt, wenn sie einen Schwellenwert für eine benutzerdefinierte Lastmetrik wie CPU- oder Speicherauslastung überschreiten.

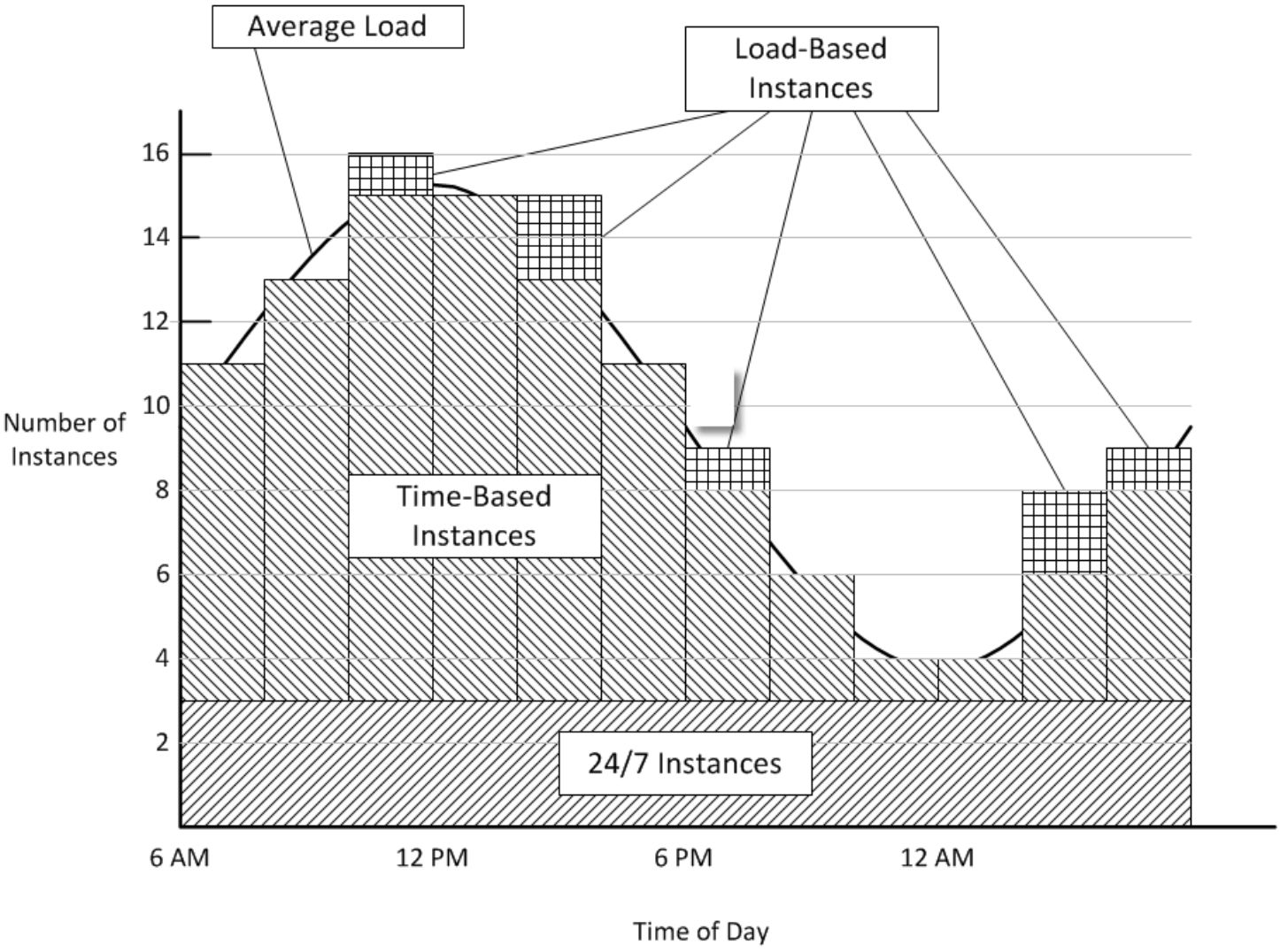
> **Note**
>
> Nachdem Sie die zeit- und lastbasierten Stack-Instances erstellt und konfiguriert haben, startet und stoppt AWS OpsWorks Stacks diese entsprechend der angegebenen Konfiguration. Sie müssen keine weiteren Änderungen vornehmen, es sei denn, Sie möchten die Konfiguration oder Anzahl der Instances ändern.

Empfehlung: Wenn Sie Stacks mit vielen Anwendungs-Instances verwalten, empfehlen wir, einen Mix aus allen drei Instance-Typen zu verwenden. Es folgt ein Beispiel zur Verwaltung einer Stack-Server-Kapazität mit folgenden Eigenschaften, um das tägliche Anforderungsvolumen zu verwalten.

- Die durchschnittliche Anforderung unterliegt am Tag einer sinusförmigen Schwankung.
- Das minimale durchschnittliche Anforderungsvolumen erfordert fünf Anwendungsserver-Instances.
- Das maximale durchschnittliche Anforderungsvolumen erfordert sechzehn Anwendungsserver-Instances.

- Die Lastspitzen des Anforderungsvolumens können in der Regel von einer oder zwei Anwendungsserver-Instances verarbeitet werden.

Dies ist ein praktisches Modell für den gegenständlichen Zweck, aber Sie können es sehr einfach an alle Schwankungen des Anforderungsvolumens anpassen und auch zum Verarbeiten von wöchentlichen Schwankungen verwenden. Das folgende Diagramm zeigt, wie Sie mit den drei Instance-Typen dieses Anforderungsvolumen verwalten können.



Dieses Beispiel hat folgende Merkmale:

- Der Stack hat drei 24/7-Instances, die ununterbrochen laufen und die Grundlast verarbeiten.
- Das Stack verfügt über 12 zeitbasierte Instances, die zum Verarbeiten der durchschnittlichen täglichen Schwankungen konfiguriert sind.

Eine läuft von 22.00 bis 2.00 Uhr, zwei weitere von 20.00 bis 22.00 Uhr und 2.00 bis 4.00 Uhr und so weiter. Der Einfachheit halber ändert das Diagramm alle zwei Stunden die Anzahl der zeitbasierten Instances, aber Sie können für eine feinere Anpassung die Anzahl stündlich ändern.

- Der Stack verfügt über ausreichend lastbasierte Instances, um Datenverkehrsspitzen zu verarbeiten, die über die Kapazität der 24/7- und zeitbasierten Instances hinausgehen.

  AWS OpsWorks Stacks startet lastbasierte Instances nur, wenn die Auslastung aller aktuell laufenden Server die angegebenen Messwerte überschreitet. Die Kosten für nicht laufende Instances sind minimal (Amazon EBS-gestützte Instances) oder gar nicht (Instances Store-Backed Instances). Es wird daher empfohlen, genügend Instances zu erstellen, um Ihr erwartetes maximales Anforderungsvolumen bequem bewältigen zu können. In diesem Beispiel sollte der Stack mindestens über drei lastbasierte Instances verfügen.

> **Note**
>
> Stellen Sie sicher, dass Sie alle drei Instance-Typen über mehrere Availability Zones verteilen, um die Auswirkungen von Service-Unterbrechungen zu minimieren.

Bewährte Methoden: Verwalten von Berechtigungen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie müssen über AWS-Anmeldeinformationen verfügen, um auf die Ressourcen Ihres Kontos zugreifen zu können. Es folgen einige allgemeine Richtlinien, anhand denen Ihren Mitarbeitern Zugriff erteilt wird.

- Zuallererst empfehlen wir, dass Sie nicht die Root-Anmeldeinformationen Ihres Kontos zum Zugreifen auf AWS-Ressourcen verwenden.

Erstellen Sie stattdessen IAM-Identitäten für Ihre Mitarbeiter und fügen Sie Berechtigungen hinzu, die den entsprechenden Zugriff ermöglichen. Jeder Mitarbeiter kann dann seine Anmeldeinformationen verwenden, um auf Ressourcen zuzugreifen.

- Mitarbeitern sollten die Berechtigungen für den Zugriff nur auf die Ressourcen erteilt werden, die sie zur Erledigung ihrer Aufgaben benötigen.

  Beispielsweise benötigen Anwendungsentwickler Zugriff nur auf die Stacks, auf denen ihre Anwendungen ausgeführt werden.

- Mitarbeitern sollten die Berechtigungen nur für die Aktionen erteilt werden, die sie zur Erledigung ihrer Aufgaben benötigen.

  Ein Anwendungsentwickler benötigt möglicherweise uneingeschränkte Berechtigungen für einen Entwicklungs-Stack sowie Berechtigungen zum Bereitstellen seiner Anwendungen für den entsprechenden Produktions-Stack. Er benötigt wahrscheinlich keine Berechtigungen zum Starten oder Stoppen der Instances auf dem Produktions-Stack oder zum Erstellen oder Löschen von Layern usw.

Weitere allgemeine Informationen zum Verwalten von Berechtigungen finden Sie unter AWS-Sicherheitsanmeldeinformationen.

Sie können AWS OpsWorks Stacks oder IAM verwenden, um Benutzerberechtigungen zu verwalten. Beachten Sie, dass die zwei Optionen sich nicht gegenseitig ausschließen. Manchmal bietet es sich an, beide Optionen zu verwenden.

AWS OpsWorks Verwaltung von Stacks-Berechtigungen

Jeder Stack hat eine Seite Permissions (Berechtigungen), auf der Sie den Benutzern die Berechtigung für den Zugriff auf den Stack erteilen und festlegen können, welche Aktionen sie durchführen können. Sie geben die Berechtigungen eines Benutzers an, indem Sie eine der folgenden Berechtigungsebenen festlegen. Jede Ebene steht für eine IAM-Richtlinie, die Berechtigungen für eine Reihe von Standardaktionen gewährt.

- Deny (Verweigern) verweigert die Berechtigung zum Interagieren mit dem Stack.
- Show (Anzeigen) gewährt die Berechtigung zum Anzeigen der Stack-Konfiguration, jedoch nicht zum Ändern des Stack-Status.
- Deploy (Bereitstellen) enthält die Show (Anzeigen)-Berechtigungen und gewährt dem Benutzer auch die Berechtigungen zum Bereitstellen von Anwendungen.

- Manage (Verwalten) enthält die Deploy (Bereitstellen)-Berechtigungen und ermöglicht es dem Benutzer auch, eine Vielzahl von Stack-Verwaltungsaktionen durchzuführen, wie z. B. Erstellen und Löschen von Instances und Ebenen.

> **Note**
>
> Die Stufe „Berechtigungen verwalten“ gewährt keine Berechtigungen für eine kleine Anzahl von AWS OpsWorks Stacks-Aktionen auf hoher Ebene, einschließlich des Erstellens oder Klonens von Stacks. Sie müssen eine IAM-Richtlinie verwenden, um diese Berechtigungen zu gewähren.

Zusätzlich zur Festlegung der Berechtigungsebenen können Sie auf der Seite Permissions (Berechtigungen) eines Stacks auch angeben, ob Benutzer SSH/RDP- oder sudo/admin-Berechtigungen für die Stack-Instances haben. Weitere Informationen zum Verwalten von AWS OpsWorks Stacks-Berechtigungen finden Sie unter Verleihen von Berechtigungen pro Stack. Weitere Informationen zum Verwalten von SSH-Zugriff finden Sie unter Verwalten des SSH-Zugriffs.

Verwaltung von IAM-Berechtigungen

Bei der IAM-Berechtigungsverwaltung verwenden Sie die IAM-Konsole, API oder CLI, um einem Benutzer eine Richtlinie im JSON-Format zuzuweisen, die seine Berechtigungen explizit spezifiziert. Weitere Informationen zur IAM-Berechtigungsverwaltung finden Sie unter Was ist IAM? .

Empfehlung: Beginnen Sie mit der Verwaltung AWS OpsWorks von Stacks Permissions. Wenn Sie die Berechtigungen eines Benutzers optimieren oder einem Benutzer Berechtigungen erteilen müssen, die nicht in den Manage (Verwalten)-Berechtigungsebenen enthalten sind, können Sie die beiden Methoden kombinieren. AWS OpsWorks Stacks bewertet dann beide Richtlinien, um die Berechtigungen des Benutzers zu ermitteln.

> **Important**
>
> Wenn ein Benutzer mehrere Richtlinien mit widersprüchlichen Berechtigungen hat, gewinnt die Ablehnung immer. Nehmen wir zum Beispiel an, dass Sie einem Benutzer eine IAM-Richtlinie zuordnen, die den Zugriff auf einen bestimmten Stack ermöglicht, dem Benutzer aber auch die Berechtigungsseite des Stacks verwenden, um dem Benutzer die

Berechtigungsstufe Verweigern zuzuweisen. Die Berechtigungsebene Deny (Verweigern) hat Vorrang, sodass der Benutzer nicht auf den Stack zugreifen kann. Weitere Informationen finden Sie unter Bewertungslogik für IAM-Richtlinien.

Angenommen, ein Benutzer soll beispielsweise die meisten Vorgänge auf einem Stack durchführen können, außer Hinzufügen oder Löschen von Layern.

- Legen Sie die Berechtigungsebene Manage (Verwalten) fest, die es dem Benutzer ermöglicht, die meisten Stack-Verwaltungsaktionen durchzuführen, z. B. Erstellen und Löschen von Ebenen.
- Fügen Sie dem Benutzer die folgende vom Kunden verwaltete Richtlinie zu, wodurch ihm die Berechtigungen zur Verwendung der DeleteLayerAktionen CreateLayerund für diesen Stack verweigert werden. Sie identifizieren den Stack anhand des Amazon-Ressourcennamens (ARN), der auf der Seite Settings (Einstellungen) des Stacks angegeben ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "opsworks:CreateLayer",
        "opsworks:DeleteLayer"
      ],
      "Resource": "arn:aws:opsworks:*:*:stack/2f18b4cb-4de5-4429-a149-ff7da9f0d8ee/"
    }
  ]
}
```

Weitere Informationen hierzu und auch zu Beispielrichtlinien finden Sie unter Verwaltung von AWS OpsWorks Stacks-Berechtigungen durch Anhängen einer IAM-Richtlinie.

Note

Eine andere Möglichkeit, die IAM-Richtlinie zu verwenden, besteht darin, eine Bedingung festzulegen, die den Stack-Zugriff auf Mitarbeiter mit einer bestimmten IP-Adresse oder einem bestimmten Adressbereich beschränkt. Um beispielsweise zu gewährleisten, dass Mitarbeiter nur innerhalb der Firewall Ihres Unternehmens auf Stacks zugreifen, legen

Sie eine Bedingung fest, die den Zugriff auf den IP-Adressbereich Ihres Unternehmens einschränkt. Weitere Informationen finden Sie unter Bedingungen.

Bewährte Methoden: Verwalten und Bereitstellen von Anwendungen und Rezeptbüchern

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks stellt Apps und Kochbücher aus einem Remote-Repository für jede neue Instanz bereit. Im Verlauf der Nutzungsdauer einer Instance müssen Sie häufig Anwendungen oder Rezeptbücher auf den Online-Instances des Stacks aktualisieren, um Funktionen, Fehlerkorrekturen oder Ähnliches hinzuzufügen. Es gibt eine Vielzahl von Möglichkeiten zum Verwalten von Stack-Anwendungen und Rezeptbüchern, aber der Ansatz sollte die folgenden allgemeinen Anforderungen erfüllen:

- Alle Instances von Produktions-Stacks sollten, mit begrenzten Ausnahmen, wie beispielsweise für A/B-Tests, denselben Anwendungscode und denselben benutzerspezifischen Rezeptbuch-Code haben.
- Das Bereitstellen eines Updates darf, auch bei Auftreten eines Fehlers, nicht den Betrieb der Website unterbrechen.

Dieser Abschnitt beschreibt empfohlene Vorgehensweisen für die Verwaltung und Bereitstellung von Anwendungen und benutzerdefinierten Rezeptbüchern.

Themen

Bewahren der Konsistenz

Im Allgemeinen müssen Sie die auf Ihren Produktions-Stacks laufenden Anwendungs- und Rezeptbuch-Codes engmaschig kontrollieren. In der Regel sollten alle Instances die aktuell genehmigte Code-Version ausführen. Ausnahmen kann es beim Aktualisieren der Anwendungen oder Rezeptbücher geben, wie unten beschrieben, sowie beim Zuweisen spezieller Fälle, wie der Durchführung von A/B-Tests.

Der Anwendungs- und Rezeptbuch-Code wird für die Instances Ihres Stacks auf zweierlei Weise von einer angegebenen Quelle bereitgestellt:

- Wenn Sie eine Instanz starten, stellt AWS OpsWorks Stacks automatisch den aktuellen App- und Kochbuchcode für die Instanz bereit.
- Für Online-Instances müssen Sie den aktuellen Code der Anwendung oder des Rezeptbuchs manuell bereitstellen, indem Sie den Bereitstellungsbefehl (für die Anwendungen) oder den Befehl "update custom cookbooks" (für Rezeptbücher) ausführen.

Da es zwei Bereitstellungsmethoden gibt, ist es wichtig, dass Sie Ihren Quellcode sorgfältig verwalten, um zu vermeiden, dass versehentlich verschiedene Codes auf verschiedenen Instances ausgeführt werden. Wenn Sie beispielsweise Apps oder Kochbücher von einem Git-Master-Branch aus bereitstellen, stellt AWS OpsWorks Stacks bereit, was sich zu diesem Zeitpunkt in diesem Branch befindet. Wenn Sie den Code im Master-Branch aktualisieren und dann eine neue Instance starten, weist diese Instance eine neuere Code-Version auf als die älteren Instances. Die neuere Version ist möglicherweise noch nicht für den Produktivbetrieb genehmigt.

Empfehlung: Amazon S3 S3-Archive

> Um sicherzustellen, dass alle Ihre Instances über die genehmigte Codeversion verfügen, empfehlen wir, Ihre Apps und Kochbücher aus einem Amazon Simple Storage Service (Amazon S3) -Archiv bereitzustellen. Dadurch wird garantiert, dass es sich bei dem Code um ein statisches Artefakt handelt — eine ZIP-Datei oder eine andere Archivdatei —, die explizit aktualisiert werden muss. Darüber hinaus ist Amazon S3 äußerst zuverlässig, sodass Sie selten, wenn überhaupt, nicht auf das Archiv zugreifen können. Um die Konsistenz weiter zu gewährleisten, sollten Sie jede Archivdatei explizit versionieren, indem Sie eine Namenskonvention oder die Amazon S3 S3-Versionierung verwenden, was einen Prüfpfad und eine einfache Möglichkeit bietet, zu einer früheren Version zurückzukehren.

Sie können beispielsweise mit einem Tool wie Jenkins eine Bereitstellungspipeline erstellen. Nachdem der Code, den Sie bereitstellen möchten, festgeschrieben und getestet wurde, erstellen Sie eine Archivdatei und laden Sie sie auf Amazon S3 hoch. Alle Bereitstellungen von Apps oder Updates von Rezeptbüchern installieren dann den Code dieser Archivdatei und jede Instance verfügt über denselben Code.

Empfehlung: Git- oder Subversion-Repositorys

Wenn Sie lieber mit einem Git oder Subversion-Repository arbeiten, stellen Sie dieses nicht aus der Master-Branch bereit. Taggen Sie stattdessen die genehmigte Version und geben Sie diese Version als Quelle für die Anwendung oder das Rezeptbuch an.

Bereitstellen von Code für Online-Instances

AWS OpsWorks Stacks stellt aktualisierten Code nicht automatisch für Online-Instances bereit. Sie müssen den Vorgang manuell auszuführen, was mit folgenden Herausforderungen verbunden ist:

- Effiziente Update-Bereitstellung, ohne die Fähigkeit der Website zu beeinträchtigen, Kundenanforderungen während des Bereitstellungsprozesses zu beantworten.
- Umgang mit einer nicht erfolgreichen Bereitstellung, sei es aufgrund von Problemen mit der bereitgestellten Anwendung oder den bereitgestellten Rezeptbüchern oder durch Probleme mit dem Bereitstellungsprozess selbst.

Der einfachste Ansatz besteht darin, den Standardbefehl "deploy" (für Anwendungen) oder den Befehl "update custom cookbooks" (für Rezeptbücher) auszuführen, wodurch das Update für jede Instance gleichzeitig bereitgestellt wird. Diese Methode ist einfach und schnell, es darf jedoch kein Fehler unterlaufen. Wenn die Bereitstellung fehlschlägt oder der aktualisierte Code Probleme aufweist, können alle Instances in Ihrem Produktions-Stack betroffen sein und Ihre Website potenziell unterbrechen oder außer Funktion setzen, bis das Problem behoben ist oder Sie auf eine frühere Version zurückgegangen sind.

Empfehlung: Verwenden Sie eine robuste Bereitstellungsstrategie, die es den Instances ermöglicht, die alte Code-Version noch so lange zur Anforderungsverarbeitung zu verwenden, bis sichergestellt ist, dass die Bereitstellung erfolgreich war und der eingehende Datenverkehr auf die neue Version umgestellt werden kann.

In den folgenden Abschnitten finden Sie zwei Beispiele für robuste Bereitstellungsstrategien und eine Besprechung, wie eine Backend-Datenbank während der Bereitstellung verwaltet werden kann.

Zugunsten einer knappen Darstellung wird die Aktualisierung von Anwendungen beschrieben, aber Sie können ähnliche Strategien auch für Rezeptbücher verwenden.

Themen

Verwenden einer fortlaufenden Bereitstellung

Eine fortlaufende Bereitstellung aktualisiert die Anwendung für die Online-Anwendungsserver-Instances eines Stacks in mehreren Phasen. Mit jeder Phase aktualisieren Sie eine Teilmenge der Online-Instances und stellen sicher, dass das Update erfolgreich ist, bevor Sie die nächste Phase starten. Wenn Probleme auftreten, können die Instances den eingehenden Datenverkehr noch mit der alten Anwendungsversion bearbeiten, bis die Probleme gelöst sind.

Das folgende Beispiel geht davon aus, dass Sie die empfohlene Methode anwenden, die Anwendungsserver-Instances Ihres Stacks über mehrere Availability Zones zu verteilen.

Ausführen einer fortlaufende Bereitstellung

1. Wählen Sie auf der Seite Deploy App (App bereitstellen) die Option Advanced (Erweitert) und anschließend eine einzelne Anwendungsserver-Instance aus und stellen Sie die Anwendung für diese Instance bereit.

   Aus Sicherheitsgründen können Sie die Instance aus dem Load Balancer entfernen, bevor Sie die Anwendung bereitstellen. Auf diese Weise ist gewährleistet, dass die Benutzer nicht auf die aktualisierte Anwendung zugreifen können, so lange nicht geprüft wurde, ob sie ordnungsgemäß funktioniert. Wenn Sie Elastic Load Balancing verwenden, entfernen Sie die Instance mithilfe der Elastic Load Balancing-Konsole, CLI oder eines SDK aus dem Load Balancer.
2. Überprüfen Sie, ob die aktualisierte Anwendung ordnungsgemäß funktioniert und die Performance-Metriken der Instance akzeptabel sind.

   Wenn Sie die Instance von einem Elastic Load Balancing Load Balancer entfernt haben, verwenden Sie die Elastic Load Balancing Balancing-Konsole, CLI oder ein SDK, um sie wiederherzustellen. Die aktualisierte Anwendungsversion verarbeitet jetzt die Benutzeranforderungen.

3. Stellen Sie das Update für die übrigen Instances in der Availability Zone bereit und prüfen Sie, ob sie korrekt arbeiten und die Metriken akzeptabel sind.
4. Wiederholen Sie Schritt 3, Zone für Zone, für die übrigen Availability Zones des Stacks. Wenn Sie besonders vorsichtig sein möchten, wiederholen Sie die Schritte 1 bis 3.

> **Note**
>
> Wenn Sie einen Elastic Load Balancing Load Balancer verwenden, können Sie dessen Integritätsprüfung verwenden, um zu überprüfen, ob die Bereitstellung erfolgreich war. Stellen Sie in diesem Fall den Ping-Pfad auf eine Anwendung ein, die Abhängigkeiten überprüft und bestätigt, dass alles ordnungsgemäß funktioniert, und nicht einfach auf eine statische Datei, die nur bestätigt, dass der Anwendungsserver ausgeführt wird.

Verwenden separater Stacks

Eine weitere Vorgehensweise zur Verwaltung von Anwendungen besteht darin, einen separaten Stack für jede Lebenszyklusphase der Anwendung zu verwenden. Die verschiedenen Stacks werden manchmal auch als Umgebungen bezeichnet. Auf diese Weise können Sie Stacks entwickeln und testen, die nicht öffentlich zugänglich sind. Wenn Sie ein Update bereitstellen möchten, leiten Sie den Datenverkehr von dem Stack, auf dem sich die aktuelle Anwendungsversion befindet, auf den Stack um, auf dem die aktualisierte Anwendungsversion gehostet wird.

Themen

- Verwenden von Entwicklungs-, Staging- und Produktions-Stacks
- Verwenden einer blau-grünen Bereitstellungsstrategie

Verwenden von Entwicklungs-, Staging- und Produktions-Stacks

Der häufigste Ansatz verwendet die folgenden Stacks.

Entwicklungs-Stack

Verwenden Sie einen Entwicklungs-Stack für Aufgaben wie die Implementierung von neuen Funktionen oder zur Fehlerbehebung. Bei einem Entwicklungs-Stack handelt es sich im Wesentlichen um einen Prototypen-Stack für den Produktivbetrieb mit den gleichen Layern, Anwendungen, Ressourcen und so weiter, die auch auf Ihrem Produktions-Stack vorhanden sind.

Da der Entwicklungs-Stack in der Regel nicht dieselbe Last bearbeiten muss wie der Stack für den Produktivbetrieb, können Sie weniger oder kleinere Instances verwenden.

Entwicklungs-Stacks sind nicht öffentlich. Sie steuern den Zugriff wie folgt:

- Beschränken Sie den Netzwerkzugriff durch Konfigurieren der Eingangsregeln für Sicherheitsgruppen des Anwendungsservers oder Load Balancers, sodass nur eingehende Anforderungen von angegebenen IP-Adressen oder Adressbereichen akzeptiert werden.

  Begrenzen Sie z.B. HTTP-, HTTPS- und SSH-Zugriffe auf Adressen in Ihrem Unternehmensadressbereich.
- Steuern Sie den Zugriff auf die AWS OpsWorks Stack-Management-Funktionen von Stacks, indem Sie die Seite „Berechtigungen“ des Stacks verwenden.

  Teilen Sie beispielsweise dem Entwicklungsteam eine Ebene zu, auf der es Berechtigungen verwalten kann und allen anderen Mitarbeiter eine Leseberechtigung.

Staging-Stack

Verwenden Sie einen Staging-Stack, um Kandidaten für einen aktualisierten Stack für den Produktivbetrieb zu testen und abzuschließen. Im Anschluss an die Entwicklung erstellen Sie einen Staging-Stack, indem Sie den Entwicklungs-Stack klonen. Führen Sie dann Ihre Test-Suite auf dem Staging-Stack aus und stellen Sie Updates bereit, um auftretende Stack-Probleme zu beheben.

Staging-Stacks sind ebenfalls nicht öffentlich. Sie steuern den Stack- und Netzwerk-Zugriff auf dieselbe Weise wie für den Entwicklungs-Stack. Beachten Sie, dass Sie beim Klonen eines Entwicklungsstapels, um einen Staging-Stack zu erstellen, die von AWS OpsWorks Stacks Permissions Management gewährten Berechtigungen klonen können. Allerdings hat das Klonen keine Auswirkungen auf Berechtigungen, die durch die IAM-Benutzerrichtlinien erteilt wurden. Zum Ändern dieser Berechtigungen müssen Sie eine IAM-Konsole, eine CLI oder ein SDK verwenden. Weitere Informationen finden Sie unter Verwalten von Benutzerberechtigungen.

Produktions-Stack

Der Produktions-Stack ist der öffentlich zugängliche Stack, der Ihre aktuelle Anwendung unterstützt. Wenn der Staging-Stack die Testphase durchlaufen hat, können Sie ihn für den Produktivbetrieb hochstufen und den alten Produktions-Stack deaktivieren. Ein Beispiel für diese Vorgehensweise finden Sie unter Verwenden einer blau-grünen Bereitstellungsstrategie.

> **Note**
>
> Anstatt die AWS OpsWorks Stacks-Konsole zum manuellen Erstellen von Stacks zu verwenden, erstellen Sie für jeden Stack eine AWS CloudFormation Vorlage. Dieser Ansatz bietet folgende Vorteile:
>
> - Geschwindigkeit und Komfort — Wenn Sie die Vorlage starten, AWS CloudFormation wird der Stack automatisch erstellt, einschließlich aller erforderlichen Instanzen.
> - Konsistenz — Speichern Sie die Vorlage für jeden Stack in Ihrem Quell-Repository, um sicherzustellen, dass Entwickler denselben Stack für denselben Zweck verwenden.

Verwenden einer blau-grünen Bereitstellungsstrategie

Die blau-grüne Bereitstellungsstrategie ist eine gängige Methode, um separate Stacks effizient zur Bereitstellung eines Anwendungs-Updates für die Produktion einzusetzen.

- Die blaue Umgebung ist der Produktions-Stack, der die aktuelle Anwendung hostet.
- Die grüne Umgebung ist der Staging-Stack, der die aktualisierte Anwendung hostet.

Wenn Sie bereit sind, die aktualisierte Anwendung zur Produktion bereitzustellen, leiten Sie das Datenaufkommen vom blauen Stack auf den grünen Stack um, der nun der neue Produktions-Stack ist. Anschließend können Sie den alten blauen Stack deaktivieren.

Das folgende Beispiel beschreibt, wie eine blaugrüne Bereitstellung mit AWS OpsWorks Stacks-Stacks in Verbindung mit Route 53 und einem Pool von Elastic Load Balancing-Load Balancing-Load Balancern durchgeführt wird. Vor dem Wechsel sollten Sie sicherstellen, dass die folgenden Schritte ausgeführt wurden:

- Das Anwendungs-Update auf dem grünen Stack hat die Tests bestanden und ist bereit für den Produktivbetrieb.
- Der grüne Stack ist identisch mit dem blauen Stack, abgesehen davon, dass er die aktualisierte Anwendung enthält und nicht öffentlich zugänglich ist.

  Beide Stacks haben dieselben Berechtigungen, die gleiche Anzahl und Art von Instances in jeder Ebene, dieselbe zeitbasierte und lastbasierte Konfiguration usw.
- Alle 24/7-Instances und geplanten, zeitbasierten Instances der grünen Stacks sind online.

- Sie verfügen über einen Pool von Elastic Load Balancing-Load Balancern, die dynamisch an eine Ebene in beiden Stacks angehängt und vorgewärmt werden können, um das erwartete Datenverkehrsvolumen zu bewältigen.
- Sie haben die Funktion für gewichtetes Routing von Route 53 verwendet, um einen Datensatz in einer Hosting-Zone zu erstellen, der Ihre gepoolten Load Balancer enthält.
- Sie haben dem Load Balancer, der an den Anwendungsserver-Layer des blauen Stacks angefügt ist, ein von null abweichendes Gewicht zugewiesen, und den nicht verwendeten Load Balancern ein Gewicht von null. Auf diese Weise wird sichergestellt, dass der Load Balancer des blauen Stacks den gesamten eingehenden Datenverkehr verarbeitet.

Umleiten der Benutzer auf den grünen Stack

1. Fügen Sie einen der ungenutzten Load Balancer des Pools an die Anwendungsserver-Ebene des grünen Stacks an. In einigen Szenarien, wie bei blitzartig auftretendem Datenverkehr oder wenn keine Lasttestkonfiguration möglich ist, um den Datenverkehr allmählich zu steigern, sollten Sie den Load Balancer vorwärmen, damit er den erwarteten Datenverkehr verarbeiten kann.
2. Nachdem alle Instances des grünen Stacks den Elastic Load Balancing Health Check bestanden haben, ändern Sie die Gewichtungen im Route 53-Datensatz, sodass der Load Balancer des grünen Stacks eine Gewichtung ungleich Null und der Load Balancer des blauen Stacks eine entsprechend reduzierte Gewichtung hat. Wir empfehlen, dass Sie zunächst den grünen Stack eine kleine Menge an Anforderungen von vielleicht 5% verarbeiten lassen und der blaue Stack den Rest verarbeitet. Sie verfügen jetzt über zwei Produktions-Stacks, wobei der grüne Stack einige der eingehenden Anforderungen verarbeitet und der blaue Stack den Rest.
3. Überwachen Sie die Performance-Metriken des grüne Stacks. Wenn sie akzeptabel sind, erhöhen Sie das Gewicht des grüne Stacks, sodass er ca. 10% des eingehenden Datenverkehrs verarbeiten kann.
4. Wiederholen Sie Schritt 3, bis der grüne Stack ca. die Hälfte des eingehenden Datenverkehrs verarbeitet. Mögliche Probleme sollten zu diesem Zeitpunkt deutlich geworden sein, sodass Sie, wenn die Performance des grünen Stacks akzeptabel ist, den Vorgang abschließen können, indem Sie das Gewicht des blauen Stacks auf null reduzieren. Der grüne Stack ist jetzt der neue blaue Stack und verarbeitet den gesamten eingehenden Datenverkehr.
5. Trennen Sie den Load Balancer aus der alten Anwendungsserver-Ebene des blauen Stacks und geben Sie ihn zurück an den Pool.

6. Obwohl der alte blaue Stack keine Benutzeranforderungen mehr verarbeitet, empfehlen wir, ihn noch eine Weile zu behalten, falls es Probleme mit dem neuen blauen Stack geben sollte. In diesem Fall können Sie das Update rückgängig machen, indem Sie den Vorgang umkehren und den eingehenden Datenverkehr auf den alten blauen Stack umleiten. Wenn Sie sicher sind, dass der neue blaue Stack akzeptabel arbeitet, setzen Sie den alten blauen Stack außer Betrieb.

Verwalten einer Backend-Datenbank

Wenn Ihre Anwendung von einer Backend-Datenbank abhängt, müssen Sie von der alten Anwendung zur neuen wechseln. AWS OpsWorks Stacks unterstützt die folgenden Datenbankoptionen.

Amazon RDS-Ebene

Mit einer Amazon Relational Database Service (Amazon RDS) -Layer erstellen Sie die RDS-DB-Instance separat und registrieren sie dann bei Ihrem Stack. Sie können eine RDS-DB-Instance immer nur jeweils mit einem Stack registrieren, aber Sie können eine RDS-DB-Instance von einem Stack auf einen anderen umschalten.

AWS OpsWorks Stacks installiert eine Datei mit den Verbindungsdaten auf Ihren Anwendungsservern in einem Format, das von Ihrer Anwendung problemlos verwendet werden kann. AWS OpsWorks Stacks fügt außerdem die Datenbankverbindungsinformationen zu den Stackkonfigurations- und Bereitstellungsattributen hinzu, auf die über Rezepte zugegriffen werden kann. Sie können Verbindungsdaten für Anwendungen auch mithilfe von JSON bereitstellen. Weitere Informationen finden Sie unter Verbinden mit einer Datenbank.

Das Aktualisieren einer Anwendung, die von einer Datenbank abhängig ist, birgt zwei grundlegende Herausforderungen:

- Sicherzustellen, dass alle Transaktionen während des Übergangs ordnungsgemäß aufgezeichnet werden und gleichzeitig Race-Bedingungen zwischen den neuen und alten Anwendungsversionen zu vermeiden.
- Den Übergang so zu gestalten, dass die Auswirkungen auf die Leistungsfähigkeit Ihrer Website begrenzt sind und Ausfallzeiten minimiert werden oder gar nicht auftreten.

Wenn Sie die in diesem Abschnitt beschriebenen Bereitstellungsstrategien verwenden, können Sie nicht einfach die Datenbank von der alten Anwendung trennen und der neuen anfügen. Beide

Anwendungsversionen werden während des Übergangs parallel ausgeführt und müssen Zugriff auf die gleichen Daten haben. Im Folgenden werden zwei Ansätze zur Übergangsverwaltung beschrieben, die beide Vorteile haben, aber auch Herausforderungen mit sich bringen.

Ansatz 1: Beide Anwendungen mit derselben Datenbank verbinden

Vorteile

- Es gibt keine Ausfallzeiten während des Übergangs.

  Eine Anwendung stoppt schrittweise den Zugriff auf die Datenbank, während die andere schrittweise übernimmt.
- Sie müssen keine Daten zwischen zwei Datenbanken synchronisieren.

Herausforderungen

- Beide Anwendungen greifen auf dieselbe Datenbank zu. Sie müssen also den Zugriff verwalten, um zu verhindern, dass Daten verloren gehen oder beschädigt werden.
- Sie müssen auf ein neues Datenbankschema migrieren, die alte Anwendungsversion muss das neue Schema verwenden können.

Wenn Sie separate Stacks verwenden, ist dieser Ansatz wahrscheinlich am besten für Amazon RDS geeignet, da die Instance nicht dauerhaft an einen bestimmten Stack gebunden ist und von Anwendungen, die auf verschiedenen Stacks ausgeführt werden, aufgerufen werden kann. Es ist jedoch nicht möglich, eine RDS-DB-Instance mit mehr als einem Stack gleichzeitig zu registrieren. Daher müssen Sie Verbindungsdaten für beide Anwendungen zur Verfügung stellen, z. B. durch die Verwendung von JSON. Weitere Informationen finden Sie unter Verwenden von benutzerdefinierten Rezepten.

Wenn Sie ein fortlaufendes Upgrade verwenden, werden die alte und die neue Anwendungsversion auf demselben Stack gehostet, sodass Sie entweder eine Amazon RDS- oder eine MySQL-Schicht verwenden können.

Ansatz 2: Jede Anwendungsversion mit einer eigenen Datenbank ausstatten

Vorteile

- Jede Version verfügt über eine eigene Datenbank, sodass die Schemata nicht kompatibel sein müssen.

Herausforderungen

- Die Daten beim Übergang zwischen den beiden Datenbanken ohne Datenverlust oder -beschädigung zu synchronisieren.

- Sicherzustellen, dass der Synchronisierungsvorgang nicht zu erheblichen Ausfallzeiten führt oder die Leistung der Website nicht erheblich beeinträchtigt wird.

Wenn Sie separate Stacks verwenden, hat jeder seine eigene Datenbank. Wenn Sie eine fortlaufende Bereitstellung ausführen, können Sie dem Stack zwei Datenbanken anfügen, eine für jede Anwendung. Wenn die alten und neuen Anwendungen kein kompatibles Datenbankschema haben, ist dieser Ansatz besser.

Empfehlung: Im Allgemeinen empfehlen wir die Verwendung einer Amazon RDS-Schicht als Backend-Datenbank einer Anwendung, da diese flexibler ist und für jedes Übergangsszenario verwendet werden kann. Weitere Informationen zum Umgang mit Übergängen finden Sie im Amazon RDS-Benutzerhandbuch.

Lokales Verpacken von Rezeptbuch-Abhängigkeiten

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können Berkshelf verwenden, um Ihre Kochbuchabhängigkeiten lokal zu verpacken, das Paket auf Amazon S3 hochzuladen und Ihren Stack so zu ändern, dass das Paket auf Amazon S3 als Kochbuchquelle verwendet wird. Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

In den folgenden Anleitungen wird beschrieben, wie Sie Ihre Kochbücher und ihre Abhängigkeiten in einer .zip-Datei vorverpacken und dann die .zip-Datei als Kochbuchquelle für Linux-Instances in Stacks verwenden. AWS OpsWorks In der ersten Anleitung wird das Verpacken eines Rezeptbuchs beschrieben. In der zweiten Anleitung wird das Verpacken mehrerer Rezeptbücher beschrieben.

Bevor Sie beginnen, installieren Sie das Chef Development Kit (auch als Chef DK bezeichnet). Dabei handelt es sich um eine Reihe von Funktionen, die von der Chef-Community erstellt wurden. Sie benötigen es, um das `chef`-Befehlszeilen-Tool verwenden zu können.

Lokales Verpacken von Abhängigkeiten in Chef 12

In Chef 12 Linux ist Berkshelf nicht mehr standardmäßig auf Stack-Instances installiert. Wir empfehlen die Installation und Verwendung von Berkshelf auf einem lokalen Entwicklungscomputer, um Ihre Rezeptbuch-Abhängigkeiten lokal zu verpacken. Laden Sie Ihr Paket einschließlich der Abhängigkeiten auf Amazon S3 hoch. Als letzten Schritt ändern Sie Ihren Chef 12 Linux-Stack so ab, dass das hochgeladene Paket als Rezeptbuchquelle verwendet wird. Beachten Sie die folgenden Unterschiede, wenn Sie Rezeptbücher in Chef 12 verpacken.

1. Erstellen Sie auf dem lokalen Computer ein Rezeptbuch, indem Sie das Befehlszeilen-Tool `chef` ausführen.

   ```
   chef generate cookbook "server-app"
   ```

   Mit diesem Befehl wird ein Rezeptbuch, ein Berksfile, eine Datei `metadata.rb` und ein Rezeptverzeichnis erstellt und in einem Ordner gespeichert, der den gleichen Namen wie das Rezeptbuch hat. Das folgende Beispiel zeigt die Struktur dessen, was erstellt wird.

   ```
   server-app <-- the cookbook you've just created
       ### Berksfile
       ### metadata.rb
       ### recipes
   ```

2. Bearbeiten Sie in einem Texteditor das Berksfile so, dass es auf Rezeptbücher zeigt, von denen das Rezeptbuch `server-app` abhängt. In unserem Beispiel möchten wir, dass `server-app` von dem Rezeptbuch `java` vom Chef Supermarket abhängt. Wir geben die Version 1.50.0 oder eine neuere Nebenversion an, Sie können aber jede veröffentlichte Version in einfachen Anführungszeichen eingeben. Speichern Sie Ihre Änderungen und schließen Sie die Datei.

   ```
   source 'https://supermarket.chef.io'
   cookbook 'java', '~> 1.50.0'
   ```

3. Bearbeiten Sie die Datei `metadata.rb`, um die Abhängigkeit hinzuzufügen. Speichern Sie Ihre Änderungen und schließen Sie die Datei.

   ```
   depends 'java' , '~> 1.50.0'
   ```

4. Wechseln Sie zu dem Rezeptbuchverzeichnis `server-app`, das Chef für Sie erstellt hat, und führen Sie dann den Befehl `package` aus, um eine `tar`-Datei des Rezeptbuchs zu erstellen.

Wenn Sie mehrere Rezeptbücher verpacken, sollten Sie diesen Befehl in dem Stammverzeichnis ausführen, in dem alle Rezeptbücher gespeichert werden. Um ein einzelnes Rezeptbuch zu verpacken, führen Sie diesen Befehl auf der Verzeichnisebene des Rezeptbuchs aus. In diesem Beispiel führen wir diesen Befehl im Verzeichnis `server-app` aus.

```
berks package cookbooks.tar.gz
```

Die Ausgabe sieht in etwa folgendermaßen aus. Die `tar.gz`-Datei wird in Ihrem lokalen Verzeichnis erstellt.

```
Cookbook(s) packaged to /Users/username/tmp/berks/cookbooks.tar.gz
```

5. Laden Sie im das Paket AWS CLI, das Sie gerade erstellt haben, auf Amazon S3 hoch. Notieren Sie den neuen URL des Rezeptbuchpakets, nachdem Sie es auf S3 hochgeladen haben. Sie benötigen diesen URL für Ihre Stack-Einstellungen.

   ```
   aws s3 cp cookbooks.tar.gz s3://bucket-name/
   ```

   Die Ausgabe sieht in etwa folgendermaßen aus.

   ```
   upload: ./cookbooks.tar.gz to s3://bucket-name/cookbooks.tar.gz
   ```

6. Ändern Sie in AWS OpsWorks Stacks Ihren Stack so, dass das Paket, das Sie hochgeladen haben, als Kochbuchquelle verwendet wird.

   a. Legen Sie die Einstellung Use custom Chef Cookbooks (Benutzerdefinierte Chef-Rezeptbücher verwenden) auf Yes (Ja) fest.

   b. Legen Sie Repository type (Repository-Typ) auf S3 Archive (S3-Archiv) fest.

   c. Fügen Sie in Repository URL (Repository-URL) die URL des Rezeptbuchpakets ein, das Sie in Schritt 5 hochgeladen haben.

   Speichern Sie die Änderungen an Ihrem Stack.

Lokales Verpacken von Abhängigkeiten für ein Rezeptbuch

1. Erstellen Sie auf dem lokalen Computer mithilfe des Chef-Befehlszeilen-Tools ein Rezeptbuch:

```
chef generate cookbook "server-app"
```

Mit diesem Befehl wird ein Rezeptbuch und eine Berksfile erstellt und in einem Verzeichnis mit dem gleichen Namen wie das Rezeptbuch gespeichert.

2. Wechseln Sie zu dem von Chef erstellten Rezeptbuch-Verzeichnis und verpacken Sie alles, indem Sie folgenden Befehl ausführen:

```
berks package cookbooks.tar.gz
```

Das Ergebnis sieht folgendermaßen aus:

```
Cookbook(s) packaged to /Users/username/tmp/berks/cookbooks.tar.gz
```

3. Laden Sie im das Paket AWS CLI, das Sie gerade erstellt haben, auf Amazon S3 hoch:

```
aws s3 cp cookbooks.tar.gz s3://bucket-name/
```

Das Ergebnis sieht folgendermaßen aus:

```
upload: ./cookbooks.tar.gz to s3://bucket-name/cookbooks.tar.gz
```

4. Ändern Sie in AWS OpsWorks Stacks Ihren Stack so, dass das Paket, das Sie hochgeladen haben, als Kochbuchquelle verwendet wird.

Lokales Verpacken von Abhängigkeiten für mehrere Rezeptbücher

In diesem Beispiel werden zwei Rezeptbücher erstellt und deren Abhängigkeiten verpackt.

1. Führen Sie auf dem lokalen Computer die folgenden `chef`-Befehle aus, um zwei Rezeptbücher zu erstellen:

```
chef generate cookbook "server-app"
chef generate cookbook "server-utils"
```

In diesem Beispiel führt das Rezeptbuch der Serveranwendung Java-Konfigurationen aus, sodass eine Java-Abhängigkeit hinzuzufügen ist.

2. Bearbeiten Sie `server-app/metadata.rb`, um eine Abhängigkeit im Community-Java-Rezeptbuch hinzuzufügen:

```
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. Weisen Sie Berkshelf das Verpacken an, indem Sie die Berksfile-Datei im Rezeptbuch-Stammverzeichnis folgendermaßen ändern:

```
source "https://supermarket.chef.io"
cookbook "server-app", path: "./server-app"
cookbook "server-utils", path: "./server-utils"
```

   Ihre Datei-Struktur sieht jetzt folgendermaßen aus:

```
..
   ### Berksfile
   ### server-app
   ### server-utils
```

4. Erstellen Sie abschließend ein Zip-Paket, laden Sie es auf Amazon S3 hoch und ändern Sie Ihren AWS OpsWorks Stacks-Stack so, dass er die neue Kochbuchquelle verwendet. Hierfür müssen Sie die Schritte 2 bis 4 in Lokales Verpacken von Abhängigkeiten für ein Rezeptbuch ausführen.

Weitere Ressourcen

Weitere Informationen zum Verpacken von Rezeptbuchabhängigkeiten finden Sie im Folgenden.

- So verpacken Sie Cookbook-Abhängigkeiten lokal mit Berkshelf im AWS-Blog DevOps
- Linux Chef 12 mit Berkshelf in den Foren AWS OpsWorks
- Berkshelf in Chef 12 in den Foren AWS OpsWorks
- Installieren von benutzerdefinierten Rezeptbüchern in diesem Handbuch
- Rezeptbuch-Repositorys in diesem Handbuch

Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Der Stack ist die Stacks-Entität der obersten Ebene. AWS OpsWorks Ein Stack steht für eine Reihe von Instances, die Sie zusammen verwalten möchten, weil sie beispielsweise einen gemeinsamen Zweck haben, z. B. für PHP-Anwendungen verwendet werden. Ein Stack fungiert nicht nur als Container, er verarbeitet auch Aufgaben, die für diese Gruppe von Instances als Ganzes gelten (z. B. Anwendungen und Rezeptbücher verwalten).

Ein Stack, auf dem hauptsächlich Webanwendungen ausgeführt werden, besteht beispielsweise etwa aus folgenden Komponenten:

- Eine Reihe von Anwendungsserver-Instances, die gemeinsam den eingehenden Datenverkehr verarbeiten
- Eine Load Balancer-Instance, die den eingehenden Datenverkehr auf die Anwendungsserver verteilt
- Eine Datenbank-Instance, die als Back-End-Datenspeicher für die Anwendungsserver dient

Üblicherweise wird für jede Umgebung ein eigener Stack generiert. Eine typische Gruppe aus Stacks besteht aus folgenden Komponenten:

- Ein Entwicklungs-Stack, auf dem Entwickler neue Funktionen hinzufügen, Fehler beheben und andere Entwicklungs- und Wartungsaufgaben ausführen können
- Ein Staging-Stack, auf dem Updates und Fixes vor der Freigabe überprüft werden
- Ein Produktions-Stack, der eingehende Anfragen von Benutzern verarbeitet

In diesem Abschnitt werden die Grundlagen der Arbeit mit Stacks beschrieben.

Themen

Migration von Stacks von Amazon EC2-Classic zu einer VPC

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Thema wird beschrieben, wie Sie einen AWS OpsWorks Stacks Stack von der Amazon EC2 Classic-Netzwerkplattform zu einem Amazon Virtual Private Cloud (Amazon VPC) -Netzwerk migrieren.

Wenn Sie Ihr AWS Konto vor dem 04.12.2013 erstellt haben, wird EC2-Classic möglicherweise in einigen Regionen unterstützt. AWS Einige Amazon EC2-Ressourcen und -Funktionen wie Enhanced Networking und neuere Instance-Typen erfordern eine Virtual Private Cloud (VPC). Einige Ressourcen können zwischen EC2-Classic und einer VPC geteilt werden, andere nicht. Um Unterbrechungen Ihres Dienstes zu vermeiden, empfehlen wir Ihnen, Ihre AWS OpsWorks Stacks Stacks auf eine VPC zu migrieren.

Themen

Voraussetzungen

Bevor Sie beginnen, müssen Sie über eine VPC verfügen, die die AWS OpsWorks Stacks Konfigurationsanforderungen erfüllt. Informationen zur Konfiguration privater Subnetze in Ihrer VPC für AWS OpsWorks Stacks finden Sie Ausführen eines Stacks in einer VPC in diesem Handbuch. Sie können mithilfe der Amazon VPC-Managementkonsole eine benutzerdefinierte VPC erstellen. Weitere Informationen finden Sie unter Konfigurationen des Amazon VPC-Konsolenassistenten und VPCs und Subnetze im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

Um mit der Migration fortzufahren, benötigen Sie die VPC-ID und die Subnetz-ID, die Sie verwenden möchten.

Migrieren Sie einen AWS OpsWorks Stacks Stack zu einer VPC

Klonen Sie zunächst einen vorhandenen EC2-Classic-Stack mithilfe der AWS OpsWorks Stacks Konsole oder API. Verschieben Sie dann die Ressourcen des vorhandenen Stacks auf den neuen Stack. Starten Sie die neuen Instanzen im geklonten Stack und stellen Sie Apps bereit. Stellen Sie sicher, dass der neue Stack funktioniert. Löschen Sie abschließend die EC2-Classic-Ressourcen aus dem EC2-Classic-Stack und anschließend den alten Stack.

1. Klonen Sie Ihren vorhandenen EC2-Classic-Stack in Ihre VPC. Beim Klonen des Stacks werden Stack-Einstellungen, Ebenen, Apps, Benutzer und Benutzerberechtigungen auf den neuen Stack kopiert. Weitere Informationen zum Klonen eines Stacks finden Sie Klonen eines Stacks in diesem Handbuch.

   Sie können einen Stack auch mithilfe der AWS OpsWorks Stacks API klonen. Wenn Sie einen Stack mithilfe der AWS SDKs AWS CLI oder klonen, legen Sie den Wert des `VpcId` Parameters auf die ID der VPC fest, in der Sie ihn erstellt haben. Voraussetzungen Weitere Informationen finden Sie unter CloneStack in der AWS OpsWorks Stacks -API-Referenz.
2. Erstellen Sie neue Instanzen in den Ebenen des geklonten Stacks. Geben Sie unbedingt die ID des Subnetzes an, in dem Sie es erstellt haben. Voraussetzungen Weitere Informationen zum Erstellen von Instanzen in einem Stack finden Sie Hinzufügen einer Instance zu einem Layer in dieser Anleitung.
3. Migrieren Sie Ihre klassischen Ressourcen wie EC2-Sicherheitsgruppen, Elastic Load Balancing Load Balancer und Elastic IP-Adressen zu Ihrer VPC und verknüpfen Sie sie dann mit dem geklonten Stack. Weitere Informationen finden Sie unter Migrieren Ihrer Ressourcen zu einer VPC im Amazon EC2 EC2-Benutzerhandbuch.

4. Registrieren Sie Amazon EBS-Volumes und Amazon RDS-Instances beim geklonten Stack. Weitere Informationen zur Registrierung von Ressourcen bei einem Stack finden Sie Registrieren von Ressourcen mit einem Stack in diesem Handbuch.

   Amazon EBS-Volumes sind keiner VPC zugeordnet, und Sie können sie instanzübergreifend sowohl in EC2-Classic-Stacks als auch in Stacks in einer VPC verwenden. Sie können Amazon RDS-Instances in EC2-Classic sowohl mit EC2-Classic-Stacks als auch mit Stacks in einer VPC registrieren.

5. Starten Sie Instances im geklonten Stack und verschieben Sie dann einen kleinen Prozentsatz Ihrer Workloads auf den geklonten Stack. Verschieben Sie beispielsweise einen kleinen Prozentsatz des Datenverkehrs auf die Elastic Load Balancing Load Balancer im geklonten Stack. Wenn Sie Amazon Route 53 verwenden, finden Sie weitere Informationen unter Weiterleiten von Datenverkehr an einen ELB-Load Balancer im Amazon Route 53-Entwicklerhandbuch.

   Leiten Sie nur einen kleinen Prozentsatz des Datenverkehrs weiter, bis Sie sicher sind, dass der neue Stack funktionsfähig ist und Ihre Anwendungen unterstützt. Lassen Sie den neuen Stack für einen Testzeitraum, z. B. eine Woche, mit einem kleinen Prozentsatz des Datenverkehrs arbeiten. Nachdem Sie sich vergewissert haben, dass der neue Stack funktioniert, leiten Sie den verbleibenden Datenverkehr an den Stack weiter.

6. Wenn Sie sicher sind, dass der geklonte Stack funktioniert, verschieben Sie den Rest Ihres Produktionsdatenverkehrs oder Ihrer Workloads auf den geklonten Stack. Sie können jetzt Instances im EC2-Classic-Stack stoppen. Wir empfehlen, den alten Stack mehrere Wochen lang verfügbar zu halten, damit Sie Workloads wieder auf den alten Stack verschieben können, falls in den Wochen nach der Migration Probleme mit dem neuen Stack auftreten.

7. Wenn der neue Stack mehrere Wochen lang funktioniert hat, löschen Sie Instances im EC2-Classic-Stack. Weitere Informationen zum Löschen von Instances finden Sie AWS OpsWorks Stacks-Instances löschen in diesem Handbuch.

   > **Important**
   >
   > Verwenden Sie nicht die Amazon EC2 EC2-Konsole oder API, um AWS OpsWorks Instances zu stoppen oder zu löschen.

8. Löschen Sie Apps im EC2-Classic-Stack. Weitere Informationen zum Löschen von Apps finden Sie unter So löschen Sie die App aus dem Stack in diesem Handbuch.

9. Löschen Sie den EC2-Classic-Stack. Weitere Informationen zum Löschen eines Stacks finden Sie Löschen eines Stacks in diesem Handbuch.

Weitere Informationen finden Sie auch unter

- Migration von EC2-Classic zu einer VPC
- Handbuch zur Fehlersuche und Fehlerbehebung
- Ausführen eines Stacks in einer VPC

Erstellen eines neuen Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Um einen neuen Stack zu erstellen, klicken Sie im AWS OpsWorks Stacks-Dashboard auf Stapel hinzufügen. Sie können dann die Seite Add Stack (Stack hinzufügen) zum Konfigurieren des Stacks verwenden. Wenn Sie fertig sind, klicken Sie auf Add Stack (Stack hinzufügen).

Themen

- Auswahl des zu erstellenden Stack-Typs
- Grundoptionen
- Erweiterte Optionen

Auswahl des zu erstellenden Stack-Typs

Bevor Sie einen Stack erstellen, müssen Sie entscheiden, welchen Stack-Typ Sie erstellen möchten. Weitere Informationen finden Sie in der folgenden Tabelle.

Für einen...	Erstellen Sie diesen Stack-Typ, wenn Sie...	Die Vorgehensweise finden Sie in diesen Anleitungen:
Beispiel-Stack	Erkunden Sie die Grundlagen von AWS OpsWorks mit einem Linux-basierten Chef 12-Stack und einer Beispiel-App Node.js.	Erste Schritte: Beispiel
Linux-basierten Chef 12-Stack	Erstellen Sie einen Linux-basierten Stack, der die neueste Version von Chef verwendet, die AWS OpsWorks unterstützt. Wählen Sie diese Option, wenn Sie ein fortgeschrittener Benutzer sind und die große Auswahl an Community-Rezeptbüchern nutzen oder Ihre eigenen Rezeptbücher schreiben möchten. Weitere Informationen finden Sie unter Chef 12 Linux.	Erste Schritte: Linux
Windows-basierten Chef-12.2 Stack	Sie einen Windows-Stack erstellen möchten.	Erste Schritte: Windows
Linux-basierten Chef 11.10-Stack	wenn Sie Chef 11.10 mit Linux und Rückwärtskompatibilität verwenden möchten.	Erste Schritte mit Chef 11 Linux-Stacks

Grundoptionen

Die Seite Add Stack (Stack hinzufügen) enthält folgende Grundoptionen.

Stack name

(Erforderlich) Ein Name, der zur Identifizierung des AWS OpsWorks Stacks in der Stacks-Konsole verwendet wird. Der Name muss nicht eindeutig sein. AWS OpsWorks Stacks generiert auch eine Stack-ID, bei der es sich um eine GUID handelt, die den Stack eindeutig identifiziert. Beispielsweise können Sie mit AWS-CLI-Befehlen wie zum Beispiel update-stack die Stack-ID zum Identifizieren des jeweiligen Stacks verwenden. Nachdem Sie einen Stack erstellt haben, können Sie seine ID finden, indem Sie Stack im Navigationsbereich und dann Stack Settings (Stack-Einstellungen) auswählen. Die ID trägt die Bezeichnung ID. OpsWorks

Region

(Erforderlich) AWS-Region, in der die Instances gestartet werden.

VPC

(Optional) ID der VPC, in die der Stack ausgeführt wird. Alle Instances werden in dieser VPC gestartet und Sie können die ID später nicht mehr ändern.

- Wenn Ihr Konto EC2-Classic unterstützt, können Sie No VPC (Keine VPC) angeben (Standardwert), wenn Sie keine VPC verwenden möchten.

  Weitere Informationen zu EC2-Classic finden Sie unter Unterstützte Plattformen.
- Wenn Ihr Konto EC2-Classic nicht unterstützt, müssen Sie eine VPC angeben.

  Die Standardeinstellung lautet Default VPC (Standard-VPC), wodurch die Benutzerfreundlichkeit von EC2-Classic mit den Vorteilen der Funktionen des VPC-Netzwerks verbunden wird. Wenn Sie Ihren Stack in einer normalen VPC ausführen möchten, müssen Sie diese mithilfe der VPC-Konsole, einer API oder einem CLI erstellen. Weitere Informationen zum Erstellen einer VPC für einen AWS OpsWorks Stacks-Stack finden Sie unter Ausführen eines Stacks in einer VPC. Allgemeine Informationen finden Sie unter Amazon Virtual Private Cloud.

Standard-Availability Zone/Standard-Subnetz

(Optional) Diese Einstellung hängt davon ab, ob Sie Ihren Stack in einer VPC erstellen:

- Wenn Ihr Konto EC2-Classic unterstützt und Sie VPC auf No VPC (Keine VPC) festlegen, wird diese Einstellung als Default Availability Zone (Standard-Availability Zone) gekennzeichnet, welche die Standard-AWS Availability Zone angibt, in der die Instances gestartet werden.
- Wenn Ihr Konto EC2-Classic nicht unterstützt oder Sie eine VPC angeben, wird dieses Feld als Default subnet (Standard-Subnetz) gekennzeichnet, wodurch das Standard-Subnetz angegeben wird, in dem die Instances gestartet werden. Sie können eine Instance in anderen

Subnetzen starten, indem Sie diesen Wert bei der Erstellung des Stacks überschreiben. Jedes Subnetz wird in einer Availability Zone zugeordnet.

Sie können AWS OpsWorks Stacks veranlassen, eine Instance in einer anderen Availability Zone oder einem anderen Subnetz zu starten, indem Sie diese Einstellung beim Erstellen der Instance überschreiben.

Weitere Informationen zum Ausführen eines Stacks in einer VPC finden Sie unter Ausführen eines Stacks in einer VPC.

Standard-Betriebssystem

(Optional) Das standardmäßig auf jeder Instance installierte Betriebssystem. Ihnen stehen folgende Optionen zur Verfügung:

- Einer der integrierten Linux-Betriebssysteme.
- Microsoft Windows Server 2012 R2.
- Ein benutzerdefiniertes AMI auf Grundlage eines unterstützten Betriebssystems.

  Wenn Sie Use custom AMI (Benutzerdefiniertes AMI verwenden) auswählen, wird das Betriebssystem durch ein benutzerdefiniertes AMI bestimmt, das Sie beim Erstellen einer Instance angeben. Weitere Informationen finden Sie unter Verwenden von benutzerdefinierten AMIs.

Weitere Informationen zu den verfügbaren Betriebssystemen finden Sie unter AWS OpsWorks Stacks-Betriebssysteme.

> **Note**
>
> Sie können das Standard-Betriebssystem beim Erstellen einer Instance überschreiben. Sie können jedoch nicht ein Linux-Betriebssystem überschreiben, um Windows anzugeben, oder Windows, um ein Linux-Betriebssystem anzugeben.

Standard-SSH-Schlüssel

(Optional) Ein Amazon EC2 EC2-Schlüsselpaar aus der Region des Stacks. Der Standardwert ist „none“. Wenn Sie ein key pair angeben, installiert AWS OpsWorks Stacks den öffentlichen Schlüssel auf der Instance.

- Für Linux-Instances können Sie den privaten Schlüssel mit einem SSH-Client verwenden, um sich bei der Instance des Stacks anzumelden.

Weitere Informationen finden Sie unter Anmelden mit SSH.

- Bei Windows-Instances können Sie den privaten Schlüssel mit der Amazon EC2 EC2-Konsole oder CLI verwenden, um das Administratorkennwort einer Instance abzurufen.

  Mit diesem Passwort wiederum können Sie sich mit einem RDP-Client als Administrator bei der Instance anmelden. Weitere Informationen finden Sie unter Anmelden mit RDP.

Weitere Informationen zum Verwalten von SSH-Schlüsseln finden Sie unter Verwalten des SSH-Zugriffs.

> **Note**
>
> Sie können diese Einstellung überschreiben, indem Sie beim Erstellen einer Instance ein anderes oder kein Schlüsselpaar angeben.

Chef-Version

Zeigt die ausgewählte Chef-Version an.

Weitere Informationen zu Chef-Versionen finden Sie unter Chef-Versionen.

Verwenden von benutzerdefinierten Chef-Rezeptbüchern

Gibt an, ob die benutzerdefinierten Chef-Rezeptbücher auf den Stack-Instances zu installieren sind.

Für Chef 12 lautet die Voreinstellung Yes (Ja). Für Chef 11 ist die Standardeinstellung Nein. Mit der Option Ja werden mehrere zusätzliche Einstellungen angezeigt, die AWS OpsWorks Stacks mit den Informationen versorgen, die es benötigt, um die benutzerdefinierten Kochbücher aus ihrem Repository auf den Instanzen des Stacks bereitzustellen, z. B. die Repository-URL. Die Details hängen davon ab, welches Repository Sie für Ihre Rezeptbücher wählen. Weitere Informationen finden Sie unter Installieren von benutzerdefinierten Rezeptbüchern.

Stack-Farbe

(Optional) Der Farbton, der zur Darstellung des Stacks auf der AWS OpsWorks Stacks-Konsole verwendet wird. Sie können verschiedene Farben zum Unterscheiden verschiedener Stacks verwenden, z. B. für Entwicklungs-, Staging- und Produktionsstacks.

Stack-Tags

Sie können auf Stack- und Layer-Ebene Tags anwenden. Wenn Sie ein Tag erstellen, wenden Sie das Tag auf alle Ressourcen innerhalb der gekennzeichneten Struktur an. Wenn Sie beispielsweise ein Tag auf einen Stack anwenden, wenden Sie das Tag auf jede Ebene und innerhalb jeder Ebene auf jede Instance, jedes Amazon EBS-Volume oder jeden Elastic Load Balancing Load Balancer in der Ebene an. Weitere Informationen dazu, wie Sie Ihre Tags aktivieren und damit die Kosten Ihrer AWS OpsWorks Stacks-Ressourcen verfolgen und verwalten können, finden Sie unter Verwenden von Kostenzuordnungs-Tags und Aktivieren von benutzerdefinierten Kostenzuordnungs-Tags im Billing and Cost Management-Benutzerhandbuch. Weitere Informationen zum Tagging in AWS OpsWorks Stacks finden Sie unter. Tags

Erweiterte Optionen

Für erweiterte Einstellungen, klicken Sie auf Advanced >> (Erweitert >>), um die Abschnitte Advanced options (Erweiterte Optionen) und Security (Sicherheit) anzuzeigen.

Der Abschnitt Advanced options (Erweiterte Optionen) enthält folgende Optionen:

Standard-Root-Gerätetyp

Legt die Art der Speicherung für das Instance-Stamm-Volume fest. Weitere Informationen hierzu finden Sie unter Speicher.

- Linux-Stacks verwenden standardmäßig ein von Amazon EBS unterstütztes Root-Volume, aber Sie können auch ein Root-Volume angeben, das vom Instance-Speicher unterstützt wird.
- Windows-Stacks müssen ein Amazon EBS-gestütztes Root-Volume verwenden.

IAM-Rolle

(Optional) Die AWS Identity and Access Management (IAM) -Rolle des Stacks, die AWS OpsWorks Stacks verwendet, um in Ihrem Namen mit AWS zu interagieren.

Standard-IAM-Instance-Profil

(Optional) Die Standard-IAM-Rolle, die den Amazon EC2 EC2-Instances des Stacks zugeordnet werden soll. Diese Rolle erteilt den auf den Stack-Instances laufenden Anwendungen Zugriffsberechtigung auf AWS-Ressourcen wie z. B. S3-Buckets.

- Um den Anwendungen spezifische Berechtigung zu erteilen, wählen Sie ein vorhandenes Instance-Profil (Rolle) mit den geeigneten Richtlinien.

- Anfänglich gewährt die Rolle des Profils keine Berechtigungen, aber Sie können die IAM-Konsole, API oder CLI verwenden, um entsprechende Richtlinien anzuhängen. Weitere Informationen finden Sie unter Festlegen von Berechtigungen für Apps auf EC2-Instances.

API-Endpunkt-Region

Dieser Einstellungswert wird von der in den Grundeinstellungen des Stacks gewählten Region übernommen. Sie können aus den folgenden regionalen Endpunkten auswählen:

- Region USA Ost (Nord-Virginia)
- Region USA Ost (Ohio)
- US West (Oregon) Region
- Region US West (N. California)
- Region Kanada (Zentral) (nur API); nicht verfügbar für Stacks, die in der AWS Management Console
- Region Asien-Pazifik (Mumbai)
- Region Asien-Pazifik (Singapur)
- Region Asien-Pazifik (Sydney)
- Region Asien-Pazifik (Tokio)
- Region Asien-Pazifik (Seoul)
- Region Europa (Frankfurt)
- Region Europa (Irland)
- Region Europa (London)
- Region Europa (Paris)
- Region Südamerika (São Paulo)

Stacks, die in einem API-Endpunkt erstellt wurden, sind nicht in einem anderen API-Endpunkt verfügbar. Da AWS OpsWorks Stacks-Benutzer auch regionsspezifisch sind, müssen Sie, wenn Sie möchten, dass AWS OpsWorks Stacks-Benutzer in einer dieser Endpunktregionen Stacks in einer anderen Endpunktregion verwalten, die Benutzer auf den Endpunkt importieren, dem die Stacks zugeordnet sind. Weitere Informationen zum Importieren von Benutzern finden Sie unter Benutzer in Stacks importieren. AWS OpsWorks

Hostname-Thema

(Optional) Eine Zeichenfolge zur Erzeugung eines standardmäßigen Hostnamens für jede Instance. Der Standardwert lautet Layer Dependent (Layer-abhängig) und verwendet den

Kurznamen des Layers der Instance und hängt jeder Instance eine eindeutige Nummer an. Beispielsweise lautet der rollenabhängige Load Balancer-Themastamm "lb". Die erste Instance, die Sie hinzufügen, erhält die Bezeichnung "lb1", die zweite "lb2" und so weiter.

OpsWorks Version des Agenten

(Optional) Ob der AWS OpsWorks Stacks-Agent automatisch aktualisiert werden soll, wenn eine neue Version verfügbar ist, oder ob eine bestimmte Agentenversion verwendet und manuell aktualisiert werden soll. Diese Funktion ist für Chef 11.10- und Chef-12-Stacks verfügbar. Die Standardeinstellung lautet Manual update (Manuelle Aktualisierung), entsprechend der neuesten Version.

AWS OpsWorks Stacks installiert auf jeder Instanz, die mit dem Service kommuniziert, einen Agenten und erledigt Aufgaben wie das Initiieren von Chef-Läufen als Reaktion auf Lebenszyklusereignisse. Dieser Agent wird regelmäßig aktualisiert. Sie verfügen über zwei Optionen, um die Agent-Version für Ihren Stack anzugeben.

- Automatisches Update — AWS OpsWorks Stacks installiert automatisch jede neue Agentenversion auf den Instanzen des Stacks, sobald das Update verfügbar ist.
- Manuelles Update — AWS OpsWorks Stacks installiert die angegebene Agentenversion auf den Instanzen des Stacks.

  AWS OpsWorks Stacks veröffentlicht eine Nachricht auf der Stack-Seite, wenn eine neue Agentenversion verfügbar ist, aktualisiert die Instanzen des Stacks jedoch nicht. Um den Agenten zu aktualisieren, müssen Sie die Stack-Einstellungen manuell aktualisieren, um eine neue Agentenversion anzugeben. AWS OpsWorks Stacks aktualisiert dann die Instanzen des Stacks.

Sie können die Standardeinstellung für die OpsWorks Agentenversion für eine bestimmte Instanz überschreiben, indem Sie deren Konfiguration aktualisieren. In diesem Fall hat die Instance-Einstellung Vorrang. Angenommen, die Standardeinstellung lautet Auto-update (Automatische Aktualisierung), sie geben jedoch für eine bestimmte Instance Manual update (Manuelle Aktualisierung) an. Wenn AWS OpsWorks Stacks eine neue Agentenversion veröffentlicht, werden automatisch alle Instanzen des Stacks aktualisiert, mit Ausnahme der Instanz, die auf Manuelles Update eingestellt ist. Um eine Agent-Version dieser Instance zu installieren, müssen Sie manuell ein Aktualisieren der Konfiguration durchführen und eine neue Version angeben.

> **Note**
>
> Die Konsole zeigt abgekürzte Agent-Versionsnummern. Um die vollständigen Versionsnummern zu sehen, rufen Sie den describe-agent-versionsAWS-CLI-Befehl oder die entsprechenden API- oder SDK-Methoden auf. Es wird die vollständige Versionsnummer der verfügbaren Agent-Versionen zurückgegeben.

Custom JSON

(Optional) Ein oder mehrere benutzerdefinierte Attribute, als JSON-Struktur formatiert. Diese Attribute sind in den Stack-Konfigurations- und Bereitstellungsattributen zusammengeführt, die auf allen Instances installiert sind und von den Rezepten verwendet werden können. Sie können ein benutzerdefiniertes JSON-Objekt verwenden, um beispielsweise die Konfigurationseinstellungen durch Überschreiben der integrierten Attribute, die die Standardeinstellungen definieren, anzupassen. Weitere Informationen finden Sie unter Nutzen eines benutzerdefinierten JSON-Objekts.

Für Sicherheit gibt es eine Option, OpsWorks Sicherheitsgruppen verwenden, mit der Sie angeben können, ob die integrierten Sicherheitsgruppen von AWS OpsWorks Stacks den Ebenen des Stacks zugeordnet werden sollen.

AWS OpsWorks Stacks bietet einen Standardsatz integrierter Sicherheitsgruppen — eine für jede Ebene —, die standardmäßig Ebenen zugeordnet sind. Mithilfe von OpsWorks Sicherheitsgruppen können Sie stattdessen Ihre eigenen benutzerdefinierten Sicherheitsgruppen bereitstellen. Weitere Informationen finden Sie unter Verwenden von Sicherheitsgruppen.

OpsWorks Sicherheitsgruppen verwenden hat die folgenden Einstellungen:

- Ja — AWS OpsWorks Stacks ordnet jeder Ebene automatisch die entsprechende integrierte Sicherheitsgruppe zu (Standardeinstellung).

  Sie können zusätzliche Sicherheitsgruppen zu einem Layer zuordnen, nachdem Sie ihn erstellt haben. Sie können jedoch nicht die integrierte Sicherheitsgruppe löschen.
- Nein — AWS OpsWorks Stacks ordnet den Ebenen keine integrierten Sicherheitsgruppen zu.

  Sie müssen geeignete EC2-Sicherheitsgruppen erstellen und allen von Ihnen erstellten Layern eine Sicherheitsgruppe zuordnen. Sie können einem Layer bei dessen Erstellung auch weiterhin die

eingebauten Sicherheitsgruppen manuell zuordnen. Benutzerdefinierte Sicherheitsgruppen sind nur für die Layer erforderlich, die benutzerdefinierte Einstellungen benötigen.

Beachten Sie Folgendes:

- Wenn OpsWorks Sicherheitsgruppen verwenden auf Ja gesetzt ist, können Sie die Portzugriffseinstellungen einer Standardsicherheitsgruppe nicht einschränken, indem Sie einer Ebene eine restriktivere Sicherheitsgruppe hinzufügen. Bei mehreren Sicherheitsgruppen verwendet Amazon EC2 die tolerantesten Einstellungen. Außerdem ist es nicht möglich, durch Änderung der Konfiguration der integrierten Sicherheitsgruppen restriktivere Einstellungen zu erstellen. Wenn Sie einen Stack erstellen, überschreibt AWS OpsWorks Stacks die Konfigurationen der integrierten Sicherheitsgruppen mit den Standardeinstellungen, sodass alle Änderungen, die Sie vornehmen, verloren gehen, wenn Sie das nächste Mal einen Stack erstellen. Wenn für eine Ebene restriktivere Sicherheitsgruppeneinstellungen erforderlich sind als für die integrierte Sicherheitsgruppe, setzen Sie OpsWorks Sicherheitsgruppen verwenden auf Nein, erstellen Sie benutzerdefinierte Sicherheitsgruppen mit Ihren bevorzugten Einstellungen und weisen Sie sie den Ebenen bei der Erstellung zu.
- Wenn Sie versehentlich eine AWS OpsWorks Stacks-Sicherheitsgruppe löschen und sie neu erstellen möchten, muss sie ein exaktes Duplikat des Originals sein, einschließlich der Groß-/Kleinschreibung des Gruppennamens. Anstatt die Gruppe manuell neu zu erstellen, empfehlen wir, AWS OpsWorks Stacks diese Aufgabe für Sie ausführen zu lassen. Erstellen Sie einfach einen neuen Stack in derselben AWS-Region — und VPC, falls vorhanden — und AWS OpsWorks Stacks erstellt automatisch alle integrierten Sicherheitsgruppen neu, einschließlich der gelöschten. Anschließend können Sie den Stack löschen, wenn Sie keine weitere Verwendung dafür haben. Die Sicherheitsgruppen bleiben erhalten.

Ausführen eines Stacks in einer VPC

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können den Benutzerzugriff auf die Instances eines Stacks mithilfe einer Virtual Private Cloud (VPC) steuern. Möglicherweise möchten Sie nicht, dass Benutzer direkt auf die Anwendungsserver oder Datenbanken Ihres Stacks zugreifen können, und möchten stattdessen sämtlichen öffentlichen Datenverkehr über einen Elastic Load Balancer leiten.

Gehen Sie wie folgt vor, um einen Stack in einer VPC auszuführen:

1. Erstellen Sie mithilfe der Amazon VPC-Konsole, API oder einer Vorlage eine AWS CloudFormation entsprechend konfigurierte VPC.
2. Geben Sie beim Erstellen des Stacks die VPC-ID an.
3. Starten Sie Stack-Instances im entsprechenden Subnetz.

Nachfolgend wird kurz erläutert, wie VPCs in AWS OpsWorks Stacks funktionieren.

Important

Wenn Sie die VPC-Endpunktfunktion verwenden, beachten Sie, dass jede Instance im Stack in der Lage sein muss, die folgenden Aktionen von Amazon Simple Storage Service (Amazon S3) aus durchzuführen:

- Installieren des Instance-Agenten
- Installieren von Ressourcen wie Ruby
- Hochladen von Chef-Protokollen
- Abrufen von Stack-Befehlen

Um diese Aktionen zu aktivieren, müssen die Stack-Instances Zugriff auf die folgenden Buckets in der Region des Stacks haben. Andernfalls schlagen die zuvor genannten Aktionen fehl.
Für Chef 12 Linux und Chef 12.2 Windows lauten die Buckets wie folgt.

Agent-Buckets	Ressourcen-Buckets	Protokoll-Buckets	DNA-Buckets
• opsworks-instance-agent-sa-Ost-1	• opsworks-instance-assets-us-Ost-2	• opsworks-us-east-2 logarithmisch	• opsworks-us-east-2-DNA

Agent-Buckets	Ressourcen-Buckets	Protokoll-Buckets	DNA-Buckets
• opsworks-instance-agent-ap-Süd-1 • opsworks-instance-agent-ap-Nordost-1 • opsworks-instance-agent-ap-Nordost-2 • opsworks-instance-agent-ap-Südost-1 • opsworks-instance-agent-ap-Südost-2 • opsworks-instance-agent-ca-zentral-1 • opsworks-instance-agent-eu-zentral-1 • opsworks-instance-agent-eu-West-1 • opsworks-instance-agent-eu-West-2 • opsworks-instance-agent-eu-West-3	• opsworks-instance-assets-us-Ost-1 • opsworks-instance-assets-ap-Süd-1 • opsworks-instance-assets-ap-Nordost-1 • opsworks-instance-assets-ap-Nordost-2 • opsworks-instance-assets-ap-Südost-1 • opsworks-instance-assets-ap-Südost-2 • opsworks-instance-assets-ca-zentral-1 • opsworks-instance-assets-eu-zentral-1 • opsworks-instance-assets-eu-West-1 • opsworks-instance-assets-eu-West-2	• opsworks-us-east-1-Protokoll • opsworks-ap-south-1-Protokoll • opsworks-ap-northeast-1-Protokoll • opsworks-ap-northeast-2 protokollieren • opsworks-ap-southeast-1-Protokoll • opsworks-ap-southeast-2 protokollieren • opsworks-ca-central-1-Protokoll • opsworks-eu-central-1-Protokoll • opsworks-eu-west-1-Protokoll • opsworks-eu-west-2 protokollieren • opsworks-eu-west-3 protokollieren	• 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

Agent-Buckets	Ressourcen-Buckets	Protokoll-Buckets	DNA-Buckets
• opsworks-instance-agent-us-Ost-1 • opsworks-instance-agent-us-Ost-2 • opsworks-instance-agent-us-West-1 • opsworks-instance-agent-us-West-2	• opsworks-instance-assets-eu-West-3 • opsworks-instance-assets-sa-Ost-1 • opsworks-instance-assets-us-West-1 • opsworks-instance-assets-us-West-2	• opsworks-sa-east-1-Protokoll • opsworks-us-west-1-Protokoll • opsworks-us-west-2 protokollieren	

Für Chef 11.10 und frühere Versionen für Linux sind dies folgende Buckets. Chef 11.4-Stacks werden auf regionalen Endpunkten außerhalb der Region USA Ost (Nord-Virginia) nicht unterstützt.

Agent-Buckets	Ressourcen-Buckets	Protokoll-Buckets	DNA-Buckets
• opsworks-instance-agent-us-Ost-2 • opsworks-instance-agent-us-Ost-1 • opsworks-instance-agent-ap-Süd-1 • opsworks-instance-agent-ap-Nordost-1 • opsworks-instance-agent-ap-Nordost-2 • opsworks-instance-agent-ap-Südost-1 • opsworks-instance-agent-ap-Südost-2 • opsworks-instance-agent-ca-zentral-1 • opsworks-instance-agent-eu-zentral-1 • opsworks-instance-agent-eu-West-1	• opsworks-instance-assets-us-Ost-2 • opsworks-instance-assets-us-Ost-1 • opsworks-instance-assets-ap-Süd-1 • opsworks-instance-assets-ap-Nordost-1 • opsworks-instance-assets-ap-Nordost-2 • opsworks-instance-assets-ap-Südost-1 • opsworks-instance-assets-ap-Südost-2 • opsworks-instance-assets-ca-zentral-1 • opsworks-instance-assets-eu-zentral-1 • opsworks-instance-assets-eu-West-1	• prod_stage-log	• prod_stage-dna

Agent-Buckets	Ressourcen-Buckets	Protokoll-Buckets	DNA-Buckets
• opsworks-instance-agent-eu-West-2 • opsworks-instance-agent-eu-West-3 • opsworks-instance-agent-us-Ost-1 • opsworks-instance-agent-us-West-1 • opsworks-instance-agent-us-West-2	• opsworks-instance-assets-eu-West-2 • opsworks-instance-assets-eu-West-3 • opsworks-instance-assets-sa-Ost-1 • opsworks-instance-assets-us-West-1 • opsworks-instance-assets-us-West-2		

Weitere Informationen finden Sie unter VPC Endpoints.

> **Note**
>
> Damit AWS OpsWorks Stacks eine Verbindung zu den VPC-Endpunkten herstellen kann, die Sie aktivieren, müssen Sie auch das Routing für Ihr NAT oder Ihre öffentliche IP konfigurieren, da der AWS OpsWorks Stacks-Agent weiterhin Zugriff auf den öffentlichen Endpunkt benötigt.

Themen

- VPC-Grundlagen
- Eine VPC für einen AWS OpsWorks Stacks-Stack erstellen

VPC-Grundlagen

Detaillierte Informationen zu VPCs finden Sie unter Amazon Virtual Private Cloud. Kurz zusammengefasst besteht eine VPC aus mindestens einem Subnetz mit je mindestens einer Instance. Jedes Subnetz ist einer Routing-Tabelle zugeordnet, über die ausgehender Datenverkehr anhand der Ziel-IP-Adresse weitergeleitet wird.

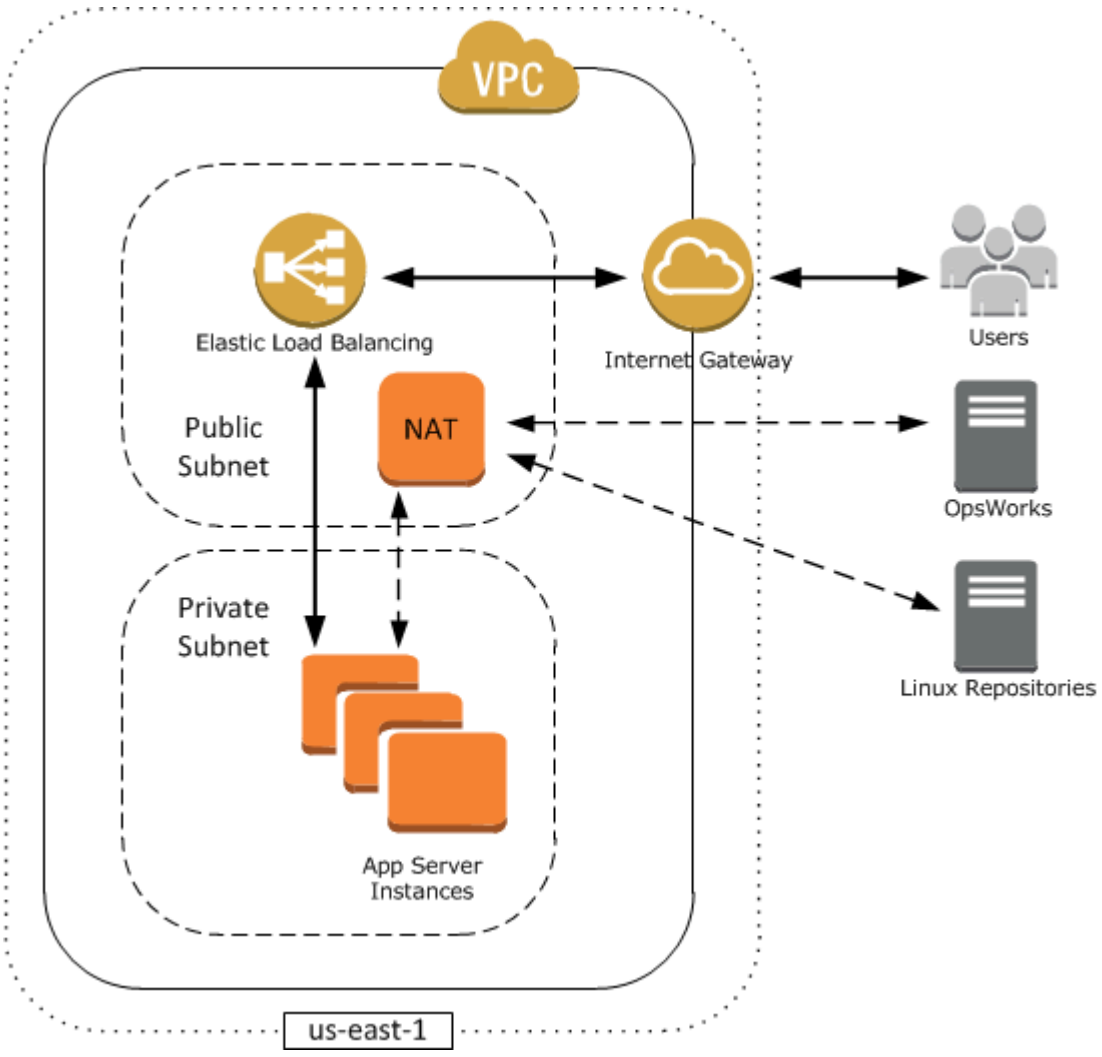
- Instances innerhalb der VPC können unabhängig vom jeweiligen Subnetz standardmäßig miteinander kommunizieren. Änderungen an Netzwerk-Zugriffskontrolllisten (ACLs), Sicherheitsgruppenrichtlinien oder die Verwendung statischer IP-Adressen können diese Kommunikation jedoch unterbrechen.
- Subnetze, deren Instances auf das Internet zugreifen können, sind öffentliche Subnetze.
- Subnetze, die mit anderen Instances in der VPC kommunizieren können, jedoch keinen Internetzugriff haben, werden als private Subnetze bezeichnet.

AWS OpsWorks Stacks erfordert, dass die VPC so konfiguriert ist, dass jede Instanz im Stack, einschließlich Instances in privaten Subnetzen, Zugriff auf die folgenden Endpunkte hat:

- Einer der AWS OpsWorks Stacks-Dienstendpunkte, die im Abschnitt „Regionssupport“ von aufgeführt sind. Erste Schritte mit AWS OpsWorks Stacks
- Einer der folgenden Instanzdienst-Endpunkte, der vom Stacks-Agenten verwendet wird. AWS OpsWorks Der Agent führt auf verwalteten Kunden-Instances ausgeführt, um Daten mit dem Service auszutauschen.
  - 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
- Alle Paket-Repositorys, die von Ihrem Betriebssystem verwendet werden, wie beispielsweise Amazon Linux- oder Ubuntu Linux-Repositorys
- Die Repositorys Ihrer App und Ihres benutzerdefinierten Rezeptbuchs

Sie können eine VPC auf unterschiedliche Weise konfigurieren, um diese Vernetzung zu schaffen. Im Folgenden finden Sie ein einfaches Beispiel dafür, wie Sie eine VPC für einen AWS OpsWorks Stacks-App-Server-Stack konfigurieren können.



Diese VPC hat mehrere Komponenten:

Subnets

Die VPC hat zwei Subnetze, ein öffentliches und ein privates.

- Das öffentliche Subnetz enthält einen Load Balancer und ein NAT-Gerät, das mit externen Adressen sowie mit Instances im privaten Subnetz kommunizieren kann.
- Das private Subnetz enthält die Anwendungsserver, die mit dem NAT und dem Load Balancer im öffentlichen Subnetz, nicht jedoch direkt mit externen Adressen kommunizieren können.

Internet-Gateway

Über den Internet-Gateway können Instances mit öffentlichen IP-Adressen wie der Load Balancer mit Adressen außerhalb der VPC kommunizieren.

Load Balancer

Der Elastic Load Balancer verteilt eingehenden Datenverkehr von Benutzern auf die Anwendungsserver im privaten Subnetz und gibt die Antworten an die Benutzer zurück.

NAT

Über das NAT-Gerät haben die Anwendungsserver begrenzten Zugriff auf das Internet. Dieser dient üblicherweise dazu, Softwareaktualisierungen von externen Repositorys herunterzuladen. Alle AWS OpsWorks Stacks-Instanzen müssen in der Lage sein, mit AWS OpsWorks Stacks und den entsprechenden Linux-Repositorys zu kommunizieren. Eine Möglichkeit, dies zu gewährleisten, besteht darin, ein NAT-Gerät mit einer entsprechenden Elastic IP-Adresse in einem öffentlichen Subnetz zu platzieren. Von dort aus können Sie ausgehenden Datenverkehr über das NAT von Instances an das private Subnetz weiterleiten.

> **Note**
>
> Eine einzelne NAT-Instance ist für den ausgehenden Datenverkehr Ihres privaten Subnetzes sehr fehleranfällig. Sie können die Zuverlässigkeit erhöhen, indem Sie zwei NAT-instances konfigurieren und so die Ausfallsicherheit erhöhen. Weitere Informationen finden Sie unter Hohe Verfügbarkeit für Amazon VPC NAT-Instances. Sie können auch einen NAT-Gateway verwenden. Weitere Informationen finden Sie unter NAT im Amazon VPC-Benutzerhandbuch.

Die optimale VPC-Konfiguration hängt von Ihrem AWS OpsWorks Stacks-Stack ab. In den nachfolgenden Beispielen sehen Sie, wann welche VPC-Konfiguration geeignet ist. Beispiele für weitere VPC-Szenarios finden Sie unter Szenarios für die Verwendung von Amazon VPC.

Working with one instance in a public subnet (Arbeiten mit einer Instance in einem öffentlichen Subnetz)

Wenn Sie über einen Einzelinstanz-Stack ohne zugehörige private Ressourcen verfügen — z. B. eine Amazon RDS-Instance, die nicht öffentlich zugänglich sein sollte —, können Sie eine VPC mit einem öffentlichen Subnetz erstellen und die Instance in dieses Subnetz stellen. Wenn Sie keine Standard-VPC verwenden, müssen Sie den Layer der Instance anweisen, der Instance eine Elastic IP-Adresse zuzuweisen. Weitere Informationen finden Sie unter OpsWorks Grundlagen der Ebene.

Working with private resources (Arbeiten mit privaten Ressourcen)

Wenn Sie über Ressourcen verfügen, die nicht öffentlich zugreifbar sein sollen, können Sie eine VPC mit einem öffentlichen und einem privaten Subnetz erstellen. In einer automatischen Skalierungsumgebung mit Lastenausgleich können Sie beispielsweise alle Amazon EC2 EC2-Instances im privaten Subnetz und den Load Balancer in einem öffentlichen Subnetz platzieren. Auf diese Weise kann nicht direkt über das Internet auf die Amazon EC2 EC2-Instances zugegriffen werden. Der gesamte eingehende Datenverkehr muss über den Load Balancer geleitet werden.

Das private Subnetz isoliert die Instances vom direkten Benutzerzugriff von Amazon EC2, sie müssen jedoch weiterhin ausgehende Anfragen an AWS und die entsprechenden Linux-Paket-Repositorys senden. Um solche Anfragen zu ermöglichen, können Sie beispielsweise ein NAT-Gerät mit eigener Elastic IP-Adresse verwenden und den ausgehenden Datenverkehr der Instances über das NAT leiten. Sie können das NAT im selben öffentlichen Subnetz wie den Load Balancer platzieren (siehe vorheriges Beispiel).

- Wenn Sie eine Back-End-Datenbank wie eine Amazon RDS-Instance verwenden, können Sie diese Instances im privaten Subnetz platzieren. Für Amazon RDS-Instances müssen Sie mindestens zwei verschiedene Subnetze in verschiedenen Availability Zones angeben.
- Wenn Sie direkten Zugriff auf Instances in einem privaten Subnetz benötigen — Sie möchten beispielsweise SSH verwenden, um sich bei einer Instance anzumelden —, können Sie einen Bastion-Host in das öffentliche Subnetz stellen, der Anfragen aus dem Internet weiterleitet.

Extending your own network into AWS (Erweitern Ihres eigenen Netzwerks in AWS)

Wenn Sie Ihr eigenes Netzwerk auf die Cloud ausweiten und aus der VPC direkt auf das Internet zugreifen möchten, können Sie einen VPN-Gateway erstellen. Weitere Informationen finden Sie unter Szenario 3: VPC mit öffentlichen und privaten Subnetzen sowie Hardware-VPN-Zugriff.

Eine VPC für einen AWS OpsWorks Stacks-Stack erstellen

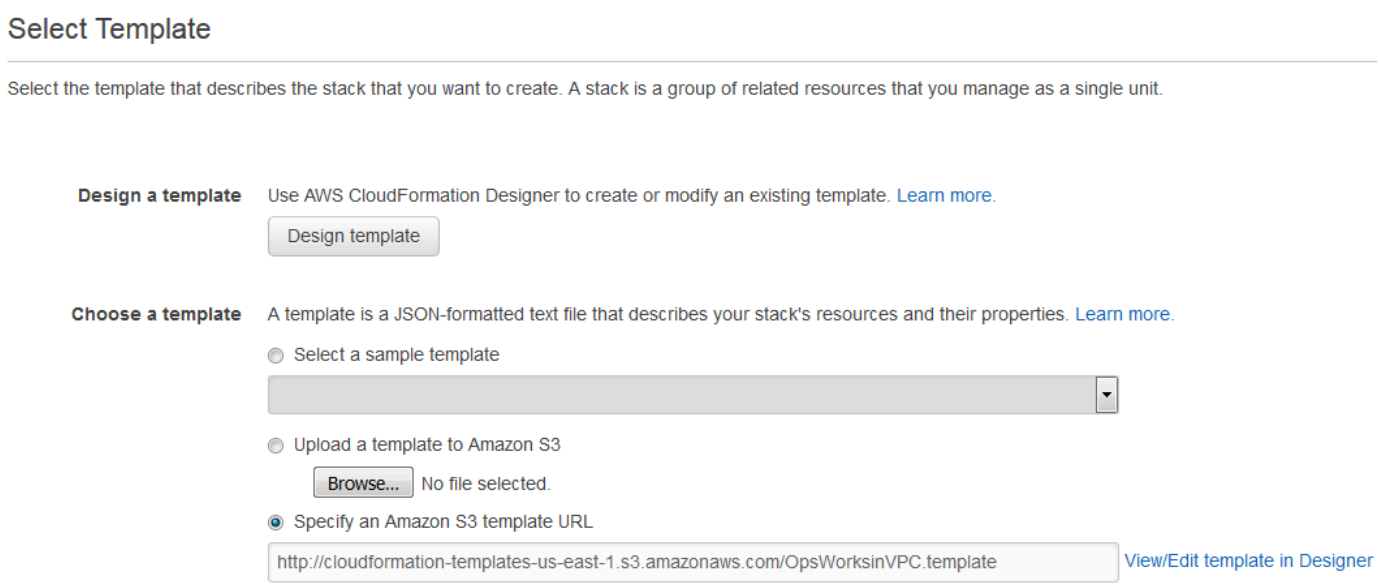
In diesem Abschnitt wird anhand einer CloudFormationAWS-Beispielvorlage gezeigt, wie Sie eine VPC für einen AWS OpsWorks Stacks-Stack erstellen. Sie können die Vorlage in der OpsWorksDatei VPCtemplates.zip herunterladen. Weitere Informationen zum manuellen Erstellen einer VPC, wie sie in diesem Thema erläutert wurde, finden Sie unter Szenario 2: VPC mit öffentlichen und privaten Subnetzen. Weitere Informationen zur Konfiguration von Routing-Tabellen, Sicherheitsgruppen usw. finden Sie in der Beispielvorlage.

> **Note**
>
> Standardmäßig zeigt AWS OpsWorks Stacks Subnetznamen an, indem deren CIDR-Bereich und Availability Zone miteinander verknüpft werden, z. B. `10.0.0.1/24 - us-east-1b` Um die Namen besser lesbar zu machen, erstellen Sie für jedes Subnetz ein Tag, wobei Key auf **`Name`** und Value auf den Subnetznamen eingestellt ist. AWS OpsWorks Stacks fügt dann den Subnetznamen an den Standardnamen an. Das private Subnetz im folgenden Beispiel hat beispielsweise ein Tag mit der Einstellung Name auf**`Private`**, das als angezeigt wird. OpsWorks `10.0.0.1/24 us-east - 1b - Private`

Sie können eine VPC-Vorlage mit nur wenigen Schritten über die AWS CloudFormation Konsole starten. Das folgende Verfahren verwendet die Beispielvorlage, um eine VPC in der Region USA Ost (Nord-Virginia) zu erstellen. Eine Anleitung dazu, wie Sie anhand der Vorlage eine VPC in anderen Regionen erstellen, finden Sie im Hinweis am Ende des Verfahrens.

So erstellen Sie die VPC

1. Öffnen Sie die AWS CloudFormation -Konsole, wählen Sie die Region US East (N. Virginia) (USA Ost (Nord-Virginia)) und anschließend Create Stack (Stack erstellen) aus.
2. Wählen Sie auf der Seite Select Template (Vorlage auswählen) Upload a template (Eine Vorlage hochladen) aus. Suchen Sie in der **`OpsWorksinVPC.template`** Datei OpsWorksVPCtemplates.zip nach der Datei, die Sie heruntergeladen haben. Wählen Sie Weiter.



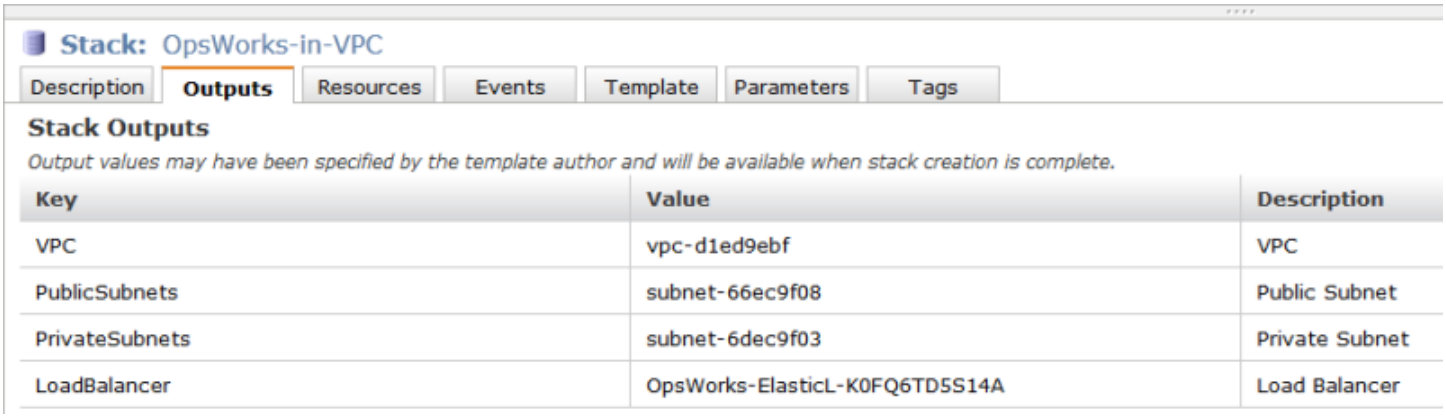
Sie können diesen Stack auch starten, indem Sie CloudFormation AWS-Beispielvorlagen öffnen, die VPC-Vorlage AWS OpsWorks Stacks suchen und Stack starten auswählen.

3. Akzeptieren Sie auf der Seite Specify Parameters (Parameter angeben) die Standardwerte und Continue (Weiter) aus.
4. Erstellen Sie auf der Seite Add Tags (Tags hinzufügen) ein Tag, legen Sie Key (Schlüssel) auf **`Name`** und Value (Wert) auf den VPC-Namen fest. Mit diesem Tag können Sie Ihre VPC leichter identifizieren, wenn Sie einen AWS OpsWorks Stacks-Stack erstellen.
5. Wählen Sie Continue (Weiter) und dann Close (Schließen) aus, um den Stack zu starten.

Hinweis: Sie können die VPC mithilfe eines der folgenden Ansätze in anderen Regionen erstellen.

- Gehen Sie zu Vorlagen in verschiedenen Regionen verwenden, wählen Sie die entsprechende Region aus, suchen Sie die VPC-Vorlage AWS OpsWorks Stacks und wählen Sie dann Stack starten aus.
- Kopieren Sie die Vorlagendatei zu Ihrem System und wählen Sie in der AWS CloudFormation -Konsole die entsprechende Region aus. Verwenden Sie im Assistenten Create Stack (Stack erstellen) die Option Upload a template to Amazon S3 (Vorlage zu Amazon S3 hochladen), um die Vorlage aus Ihrem System hochzuladen.

Die Beispielvorlage enthält Ausgaben, die die VPC-, Subnetz- und Load Balancer-IDs bereitstellen, die Sie zum Erstellen des AWS OpsWorks Stacks-Stacks benötigen. Sie können sie sehen, indem Sie unten im Konsolenfenster auf die Registerkarte Ausgaben klicken. AWS CloudFormation



Aktualisieren eines Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können die Konfiguration eines Stacks auch nach dem Erstellen jederzeit anpassen. Klicken Sie auf der Seite Stack auf Stack Settings (Stack-Einstellungen) und anschließend auf Edit (Bearbeiten), um die Seite Settings (Einstellungen) anzuzeigen. Nehmen Sie die gewünschten Änderungen vor und klicken Sie auf Save (Speichern).

Die Einstellungen entsprechen den unter Erstellen eines neuen Stacks vorgestellten Einstellungen. Weitere Informationen finden Sie in diesem Thema. Beachten Sie jedoch Folgendes:

- Sie können die Region- oder VPC-ID nicht ändern.
- Wenn Ihr Stack in einer VPC ausgeführt wird, verfügt er über die Einstellung Default subnet (Standard-Subnetz), über die die Subnetze der VPC angezeigt werden. Wenn Ihr Stack nicht in einer VPC ausgeführt wird, heißt diese Einstellung Default Availability Zones (Standard-Availability Zones) und zeigt die Availability Zones der Region an.
- Sie können das Standardbetriebssystem ändern, es ist jedoch nicht möglich, ein Linux-Betriebssystem für einen Windows-Stack auszuwählen oder umgekehrt.

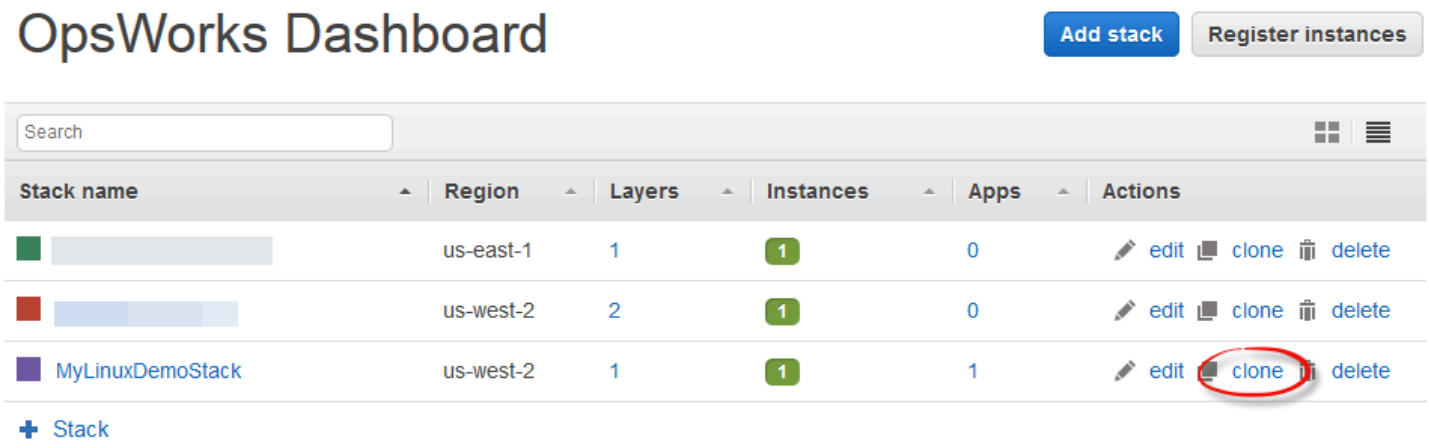
- Wenn Sie Standardeinstellungen für eine Instance wie Hostname theme (Hostname-Thema) oder Default SSH key (Standard-SSH-Schlüssel) ändern, werden diese Werte nur für neu erstellte Instances übernommen. Bereits bestehende Instances bleiben unverändert.
- Eine Änderung des Namens ändert den Namen, der von der Konsole angezeigt wird. Der zugrunde liegende Kurzname, den Stacks zur Identifizierung des AWS OpsWorks Stacks verwendet, wird dadurch nicht geändert.
- Bevor Sie die Einstellung OpsWorks Sicherheitsgruppen verwenden von Ja in Nein ändern, muss jede Ebene zusätzlich zur integrierten Sicherheitsgruppe der Ebene über mindestens eine Sicherheitsgruppe verfügen. Weitere Informationen finden Sie unter Bearbeiten der Konfiguration einer Ebene OpsWorks .

  AWS OpsWorks Stacks löscht dann die integrierten Sicherheitsgruppen aus jeder Ebene.
- Wenn Sie „ OpsWorks Sicherheitsgruppen verwenden“ von „Nein“ in „Ja“ ändern, fügt AWS OpsWorks Stacks jeder Ebene die entsprechende integrierte Sicherheitsgruppe hinzu, löscht jedoch nicht die vorhandenen Sicherheitsgruppen.

Klonen eines Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Es kann hilfreich sein, mehrere Kopien eines Stacks zu erstellen. Möglicherweise möchten Sie zur Notfallwiederherstellung oder als Präventionsmaßnahme ein redundantes System erstellen oder ein vorhandener Stack soll als Ausgangspunkt für einen neuen Stack dienen. Die einfachste Möglichkeit ist es, den ursprünglichen Stack zu klonen. Wählen Sie im AWS OpsWorks Stacks-Dashboard in der Spalte Aktionen der Zeile für den Stack, den Sie klonen möchten, die Option Klonen aus, wodurch die Seite „Stack klonen“ geöffnet wird.



Die Einstellungen für den geklonten Stack sind zunächst mit denen des ursprünglichen Stacks identisch. Der einzige Unterschied besteht darin, dass an den Namen des neuen Stacks "copy" angehängt wird. Weitere Informationen zu diesen Einstellungen finden Sie unter Erstellen eines neuen Stacks. Darüber hinaus gibt es zwei weitere, optionale Einstellungen:

Berechtigungen

Wenn all permissions (alle Berechtigungen) (Standard) ausgewählt ist, werden die Berechtigungen des ursprünglichen Stacks auf den geklonten Stack übertragen.

Apps

Listet die Apps auf, die auf dem ursprünglichen Stack bereitgestellt wurden. Wenn das Kontrollkästchen neben einer App aktiviert ist (Standard), wird die App auch auf dem geklonten Stack bereitgestellt.

> **Note**
>
> Sie können einen Stack nicht von einem regionalen Endpunkt auf einen anderen klonen. Sie können beispielsweise keinen Stack von der Region USA West (Oregon) (us-west-2) in die Region Asien-Pazifik (Mumbai) (ap-south-1) klonen.

Wenn Sie die Einstellungen abgeschlossen haben, wählen Sie Stack klonen. AWS OpsWorks Stacks erstellt einen neuen Stack, der aus den Ebenen des Quellstapels und optional aus seinen Apps und Berechtigungen besteht. Die Layer haben dieselbe Konfiguration wie die ursprünglichen Layer, abgesehen von den Änderungen, die Sie ausgeführt haben. Durch das Klonen werden jedoch keine Instances erstellt. Sie müssen für jeden Layer des geklonten Stacks erst die erforderlichen Instances

erstellen und starten. Wie bei jedem Stack können Sie auf dem geklonten Stack die normalen Verwaltungsaufgaben wie Hinzufügen, Löschen oder Bearbeiten von Layern sowie Hinzufügen und Bereitstellen von Apps durchführen.

Um den geklonten Stack betriebsbereit zu machen, starten Sie die Instanzen. AWS OpsWorks Stacks richtet jede Instanz entsprechend ihrer Layer-Mitgliedschaft ein und konfiguriert sie. Außerdem werden alle Anwendungen wie bei einem neuen Stack bereitgestellt.

Führen Sie AWS OpsWorks Stacks Stack-Befehle aus

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks bietet eine Reihe von Stack-Befehlen, mit denen Sie eine Vielzahl von Operationen auf den Instanzen eines Stacks ausführen können. Um einen Stack-Befehl auszuwählen, klicken Sie auf Run Command (Befehl ausführen) auf der Seite Stack. Anschließend wählen Sie den entsprechenden Befehl aus, geben bei Bedarf Optionen an und drücken auf die Schaltfläche unten rechts, die den Namen des Befehls anzeigt.

> **Note**
>
> AWS OpsWorks Stacks unterstützt auch eine Reihe von Bereitstellungsbefehlen, mit denen Sie die Anwendungsbereitstellung verwalten können. Weitere Informationen finden Sie unter Bereitstellen von Anwendungen.

Sie können die folgenden Stack-Befehle auf einem beliebigen Stack ausführen.

Aktualisieren benutzerdefinierter Rezeptbücher

Aktualisiert die benutzerdefinierten Rezeptbücher der Instances mit der aktuellen Version aus dem Repository. Mit diesem Befehl können Sie keine Rezepte ausführen. Zum Ausführen

der aktualisierten Rezepte können Sie den Stack-Befehl `Execute Recipes`, `Setup` oder `Configure` verwenden oder die Anwendung erneut bereitstellen, um die Bereitstellungsrezepte auszuführen. Weitere Informationen zu benutzerdefinierten Rezeptbüchern finden Sie unter Cookbooks und Rezepte.

Ausführen von Rezepten

Führt eine bestimmte Gruppe von Rezepten auf den Instances aus. Weitere Informationen finden Sie unter Manuelles Ausführen von Rezepten.

Aufstellen

Führt die Einrichtungsrezepte der Instances aus.

Konfiguration

Führt die Konfigurationsrezepte der Instances aus.

> **Note**
>
> Wenn Sie Setup (Einrichten) oder Configure (Konfigurieren) verwenden möchten, um die Rezepte auf einer Instance auszuführen, müssen die Rezepte dem entsprechenden Lebenszyklusereignis des Instance-Layers zugeordnet sein. Weitere Informationen finden Sie unter Ausführen von Rezepten.

Sie können die folgenden Stack-Befehle nur auf Linux-basierten Stacks ausführen.

Installieren von Abhängigkeiten

Installiert die Pakete der Instance. Ab Chef 12 ist dieser Befehl nicht mehr verfügbar.

Aktualisieren von Abhängigkeiten

(Nur Linux. Ab Chef 12 ist dieser Befehl nicht verfügbar.) Installiert regelmäßige Betriebssystemaktualisierungen und Paketaktualisierungen. Die Details sind vom Betriebssystem der Instances abhängig. Weitere Informationen finden Sie unter Verwalten von Sicherheitsupdates.

Mit dem Befehl Upgrade Operating System (Betriebssystem aktualisieren) können Sie Instances auf eine neue Amazon Linux-Version aktualisieren.

Aktualisieren des Betriebssystems

(Nur Linux.) Aktualisiert die Amazon Linux-Betriebssysteme der Instances auf die neueste Version. Weitere Informationen finden Sie unter AWS OpsWorks Stacks-Betriebssysteme.

> **Important**
>
> Im Anschluss an die Ausführung von Upgrade Operating System (Betriebssystem aktualisieren) sollten Sie auch Setup (Einrichten) ausführen. Auf diese Weise wird sichergestellt, dass die Services ordnungsgemäß neu gestartet werden.

Stack-Befehle verfügen über die folgenden Optionen, von denen einige nur für bestimmte Befehle angezeigt werden.

Kommentar

(Optional) Geben Sie benutzerdefinierte Bemerkungen ein, die für Sie wichtig sind.

Auszuführende Rezepte

(Pflichtfeld) Diese Einstellung wird nur angezeigt, wenn Sie den Befehl Execute Recipes (Rezepte ausführen) auswählen. Geben Sie die auszuführenden Rezepte ein und verwenden Sie dafür das Standardformat *cookbook_name*::*recipe_name*, getrennt durch Kommas. Wenn Sie mehrere Rezepte angeben, führt AWS OpsWorks Stacks sie in der aufgeführten Reihenfolge aus.

Neustart zulassen

(Optional) Diese Einstellung wird nur angezeigt, wenn Sie den Befehl Upgrade Operating System (Upgrade des Betriebssystems) auswählen. Der Standardwert ist Yes, wodurch AWS OpsWorks Stacks angewiesen wird, die Instanzen nach der Installation des Upgrades neu zu starten.

Benutzerdefinierte JSON-Chef-Dateien

(Optional) Wählen Sie Advanced (Erweitert) zum Anzeigen dieser Option, mit der Sie benutzerdefinierte JSON-Attribute in Stack-Konfiguration und Bereitstellung von Attributen integrieren können.

Instances

(Optional) Geben Sie die Instances an, auf denen der Befehl ausgeführt werden soll. Standardmäßig sind alle Online-Instances ausgewählt. Wenn Sie den Befehls auf einer Teilmenge der Instances ausführen möchten, wählen Sie die entsprechenden Layers oder Instances aus.

> **Note**
>
> Möglicherweise finden Sie Ausführungen für execute_recipes, die Sie nicht aufgeführt hatten, auf den Seiten Deployment (Bereitstellung) und Commands (Befehle). Dies ist in der Regel das Ergebnis einer Berechtigungsänderung, z.B. bei Zuweisung oder Entfernen von SSH-Berechtigungen für einen Benutzer. Wenn Sie eine solche Änderung vornehmen, verwendet AWS OpsWorks Stacks execute_recipes, um die Berechtigungen für die Instanzen zu aktualisieren.

Nutzen eines benutzerdefinierten JSON-Objekts

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Mit verschiedenen AWS OpsWorks Stacks-Aktionen können Sie benutzerdefiniertes JSON angeben, das AWS OpsWorks Stacks auf Instanzen installiert und von Rezepten verwendet werden kann.

Sie können in folgenden Situationen ein benutzerdefiniertes JSON-Objekt festlegen:

- Beim Erstellen, Aktualisieren oder Klonen eines Stacks

  AWS OpsWorks Stacks installiert das benutzerdefinierte JSON auf allen Instances für alle nachfolgenden Lebenszyklusereignisse.
- Beim Ausführen eines Bereitstellungs- oder Stack-Befehls

  AWS OpsWorks Stacks gibt das benutzerdefinierte JSON nur für dieses Ereignis an Instances weiter.

Das benutzerdefinierte JSON-Objekt muss durch ein korrekt formatiertes, gültiges JSON-Objekt dargestellt werden. Beispielsweise:

```
{
  "att1": "value1",
  "att2": "value2"
  ...
}
```

AWS OpsWorks Stacks speichert benutzerdefiniertes JSON an den folgenden Orten:

Auf Linux-Instances:

- `/var/chef/runs/run-ID/attribs.json`
- `/var/chef/runs/run-ID/nodes/hostname.json`

Auf Windows-Instances:

- `drive:\chef\runs\run-ID\attribs.json`
- `drive:\chef\runs\run-ID\nodes\hostname.json`

> **Note**
>
> Auf Chef 11.10 und früheren Versionen für Linux wird das benutzerdefinierte JSON-Objekt auf Linux-Instances im folgenden Pfad gespeichert. Windows-Instances sind nicht verfügbar und es gibt keine Datei `attribs.json`. Die Protokolle werden im selben Verzeichnis gespeichert wie die JSON. Weitere Informationen zu benutzerdefinierter JSON in Chef 11.10 und früheren Versionen für Linux finden Sie unter Overriding Attributes with Custom JSON (Überschreiben von Attributen mit einem benutzerdefinierten JSON-Objekt) und Chef Logs (Chef-Protokolle).
> `/var/lib/aws/opsworks/chef/hostname.json`

In den genannten Pfaden ist *run-ID* eine eindeutige ID, die AWS OpsWorks Stacks jeder Chef-Ausführung auf einer Instance zuweist. *hostname (Host-Name)* ist der Hostname der Instance.

Verwenden Sie die Standard-Chefnode-Syntax, um mittels Chef-Rezepten auf das benutzerdefinierte JSON-Objekt zuzugreifen.

Angenommen, Sie möchten einfache Einstellungen für eine App definieren, die Sie bereitstellen möchten, beispielsweise ob die App direkt sichtbar ist und welche Vorder- und Hintergrundfarben für die App verwendet werden sollen. Sie können diese App-Einstellungen mit einem JSON-Objekt wie folgt definieren:

```
{
  "state": "visible",
  "colors": {
    "foreground": "light-blue",
    "background": "dark-gray"
  }
}
```

So deklarieren Sie das benutzerdefinierte JSON-Objekt für einen Stack:

1. Wählen Sie auf der Stack-Seite Stack Settings (Stack-Einstellungen) und dann Edit (Bearbeiten) aus.
2. Geben Sie unter Custom Chef JSON (Benutzerdefinierte JSON-Chef-Dateien) das JSON-Objekt ein und wählen Sie dann Save (Speichern) aus.

> **Note**
>
> Sie können benutzerdefinierte JSON-Objekte auf Bereitstellungs-, Layer- und Stacks-Ebene deklarieren. Dies kann nützlich sein, wenn Sie ein benutzerdefiniertes JSON-Objekt nur für einzelne Bereitstellungen oder Layer bereitstellen möchten. Oder Sie möchten das auf Stack-Ebene deklarierte benutzerdefinierte JSON-Objekt temporär mit einer benutzerdefinierten JSON auf Layer-Ebene überschreiben. Wenn Sie benutzerdefinierte JSON-Objekte auf mehreren Ebenen deklarieren, werden die auf Layer- und Stack-Ebene deklarierten benutzerdefinierten JSON-Objekte von dem auf der Bereitstellungsebene deklarierten benutzerdefinierten JSON-Objekt überschrieben. Benutzerdefiniertes JSON-Objekt, das nur auf Stack-Ebene deklariert ist, wird von jeglichen, nur auf Layer-Ebene deklarierten benutzerdefinierten JSON-Objekten überschrieben.
> Um mithilfe der AWS OpsWorks Stacks-Konsole benutzerdefiniertes JSON für eine Bereitstellung anzugeben, wählen Sie auf der Seite App bereitstellen die Option Erweitert aus. Geben Sie das benutzerdefinierte JSON-Objekt in das Feld Custom Chef JSON (Benutzerdefiniertes Chef JSON) ein und wählen Sie dann Save (Speichern) aus.
> Um mit der AWS OpsWorks Stacks-Konsole benutzerdefiniertes JSON für eine Ebene anzugeben, wählen Sie auf der Seite „Ebenen“ die Option Einstellungen für die gewünschte

Ebene aus. Geben Sie das benutzerdefinierte JSON-Objekt in das Feld Custom JSON (Benutzerdefiniertes JSON-Objekt) ein und wählen Sie dann Save (Speichern) aus. Weitere Informationen finden Sie unter Bearbeiten der Konfiguration einer Ebene OpsWorks und Bereitstellen von Anwendungen.

Wenn Sie einen Bereitstellungs- oder Stack-Befehl ausführen, können Rezepte diese benutzerdefinierten Werte unter Verwendung der Standard-Chef-node-Syntax abrufen, über die die Werte direkt auf die Hierarchie des benutzerdefinierten JSON-Objekts abgebildet werden. Das folgende Rezept schreibt beispielsweise Meldungen zu den genannten benutzerdefinierten JSON-Werten in das Chef-Protokoll:

```
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']}' **********")
```

Dieser Ansatz kann nützlich sein, um Daten an Rezepte zu übergeben. AWS OpsWorks Stacks fügt diese Daten der Instanz hinzu, und Rezepte können die Daten mithilfe der node Standard-Chef-Syntax abrufen.

Note

Ein benutzerdefiniertes JSON-Format ist auf 120 KB begrenzt. Wenn Sie mehr Kapazität benötigen, empfehlen wir, einige Daten auf Amazon Simple Storage Service (Amazon S3) zu speichern. Ihre benutzerdefinierten Rezepte können dann die AWS-CLI oder die verwenden AWS SDK for Ruby, um die Daten aus dem Amazon S3 S3-Bucket auf Ihre Instance herunterzuladen.

Löschen eines Stacks

Important

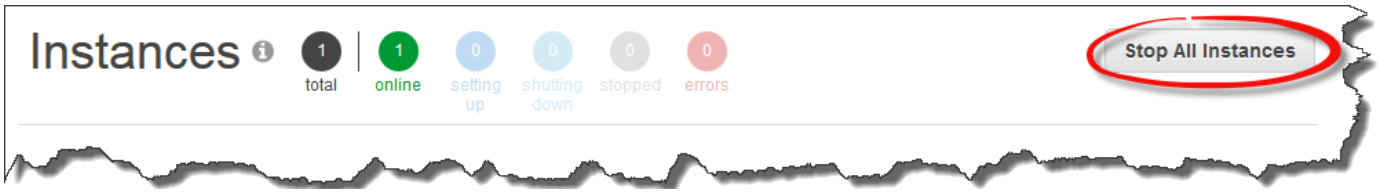
Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

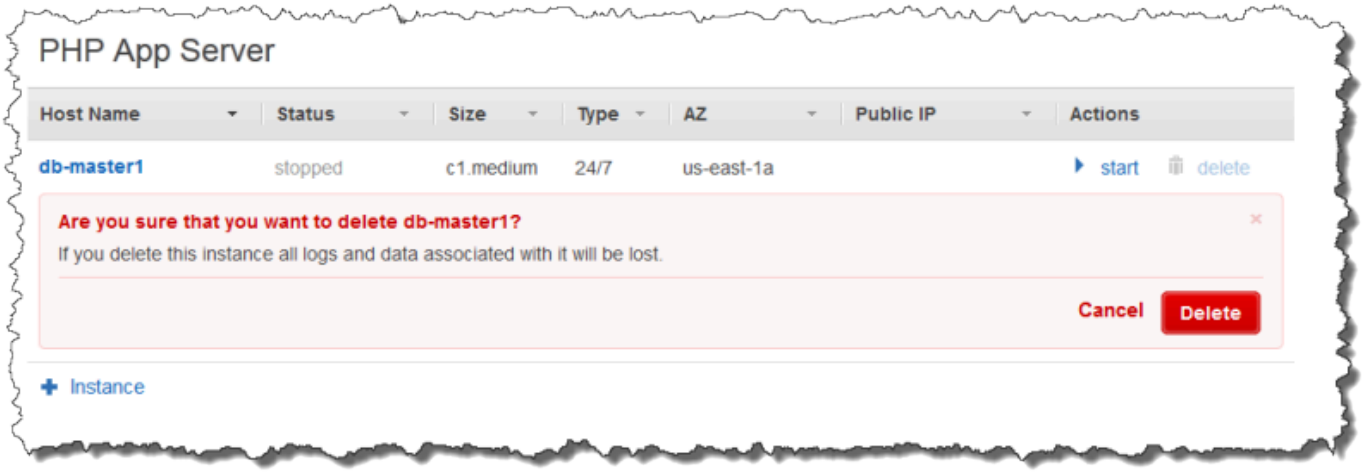
Wenn Sie einen Stack nicht mehr benötigen, können Sie ihn löschen. Nur leere Stacks können gelöscht werden. Sie müssen zuerst alle Instances, Apps und Layer im Stack löschen.

So löschen Sie einen Stack

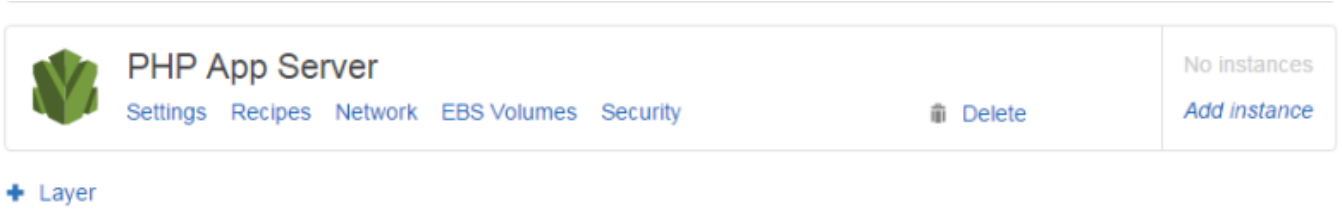
1. Wählen Sie im AWS OpsWorks Stacks-Dashboard den Stack aus, den Sie herunterfahren und löschen möchten.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie auf der Seite Instances Stop all Instances (Alle Instances anhalten) aus.



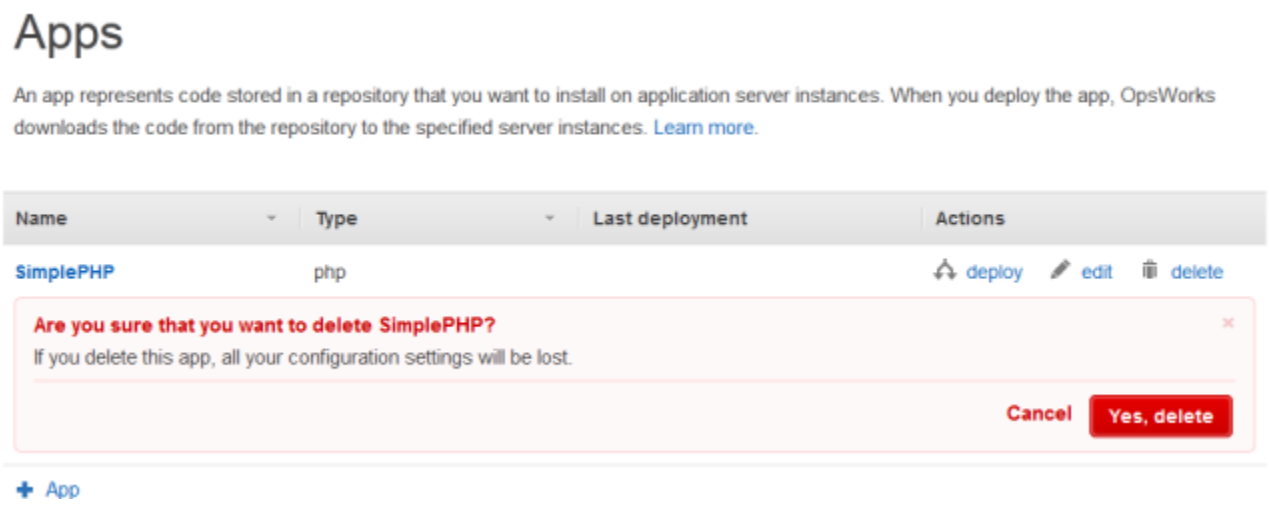
4. Nachdem die Instanzen gestoppt wurden, wählen Sie für jede Instanz im Layer in der Spalte Aktionen die Option Löschen aus. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Yes, Delete (Ja, löschen) aus.



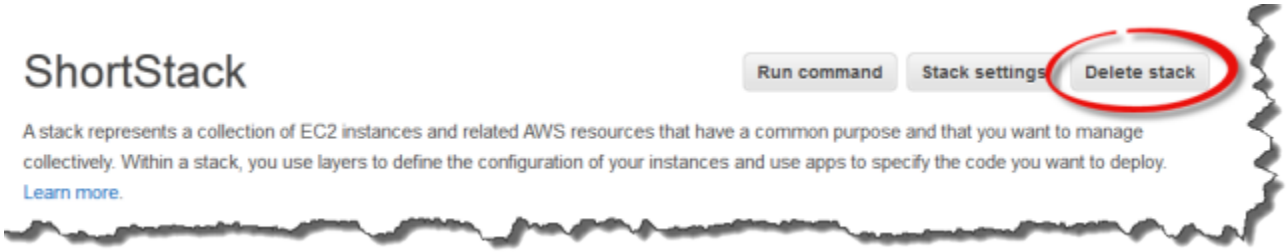
5. Nachdem alle Instances gelöscht wurden, wählen Sie im Navigationsbereich Layers aus.
6. Wählen Sie auf der Seite Layers für jede Ebene im Stack delete (Löschen) aus. Wählen Sie bei der Bestätigungsanfrage Yes, Delete (Ja, löschen) aus.



7. Nachdem alle Layer gelöscht wurden, wählen Sie im Navigationsbereich Apps aus.
8. Wählen Sie auf der Apps-Seite für jede App im Stack in der Spalte Aktionen die Option Löschen aus. Wählen Sie bei der Bestätigungsanfrage Yes, Delete (Ja, löschen) aus.



9. Nachdem alle Apps gelöscht wurden, wählen Sie im Navigationsbereich Stack aus.
10. Wählen Sie auf der Stacks-Seite Delete Stack (Stack löschen) aus. Wählen Sie bei der Bestätigungsanfrage Yes, Delete (Ja, löschen) aus.



Löschen anderer AWS Ressourcen, die von einem Stack verwendet werden

Sie verwenden andere AWS Ressourcen mit AWS OpsWorks Stacks, um Ihre Stacks zu erstellen und zu verwalten. Wenn Sie einen Stapel löschen, sollten Sie in Erwägung ziehen, auch Ressourcen zu löschen, die mit dem Stack gearbeitet haben, wenn ein anderer Stack sie nicht verwendet und Ressourcen außerhalb von AWS OpsWorks Stacks sie nicht verwenden. Im Folgenden werden

Gründe für die Bereinigung externer AWS Ressourcen, die Sie mit einem Stack verwendet haben, vorgeschlagen.

- Für externe AWS Ressourcen können weiterhin Gebühren auf Ihrem AWS Konto anfallen.
- Ressourcen wie Amazon S3 S3-Buckets können personenbezogene, sensible oder vertrauliche Informationen enthalten.

> **Important**
>
> Löschen Sie diese Ressourcen nicht, wenn sie von anderen Stacks verwendet werden. Beachten Sie, dass IAM-Rollen und Sicherheitsgruppen global sind, sodass Stacks in anderen Regionen möglicherweise dieselben Ressourcen verwenden.

Im Folgenden finden Sie weitere AWS Ressourcen, die Stacks verwenden, sowie Links zu Informationen darüber, wie Sie sie löschen können.

Servicerollen und Instance-Profile

Wenn Sie einen Stack erstellen, geben Sie eine IAM-Rolle und ein Instanzprofil an, das AWS OpsWorks Stacks verwendet, um zulässige Ressourcen in Ihrem Namen zu erstellen. AWS OpsWorks erstellt die Rolle und das Instanzprofil für Sie, falls Sie keine vorhandenen auswählen. Die Rollen- und Instanzprofile, die für Sie AWS OpsWorks erstellt werden`aws-opsworks-ec2-role`, haben jeweils den Namen `aws-opsworks-service-role` und. Wenn keine anderen Stacks in Ihrem Konto die IAM-Rolle und das Instanzprofil verwenden, können Sie diese Ressourcen problemlos löschen. Informationen zum Löschen von IAM-Rollen und Instanzprofilen finden Sie unter Löschen von Rollen oder Instanzprofilen im IAM-Benutzerhandbuch.

Sicherheitsgruppen

In AWS OpsWorks Stacks können Sie benutzerdefinierte Sicherheitsgruppen auf Layer-Ebene angeben. Sie erstellen Sicherheitsgruppen mithilfe der Amazon EC2 EC2-Konsole oder API. Stacks und Layer in anderen Regionen können dieselben Sicherheitsgruppen verwenden, weil Sicherheitsgruppen global sind. Sie können eine Sicherheitsgruppe löschen, wenn sie nicht von anderen AWS Ressourcen verwendet wird. Weitere Informationen zum Löschen einer Sicherheitsgruppe finden Sie unter Löschen einer Sicherheitsgruppe im Amazon EC2 EC2-Benutzerhandbuch.

Amazon-EBS-Volumes

In AWS OpsWorks Stacks fügen Sie EBS-Volumes auf Layer-Ebene hinzu, und sie werden an Instances auf der Ebene angehängt. Sie erstellen EBS-Volumes mithilfe der Amazon EC2-Servicekonsole oder API und hängen sie dann auf Layer-Ebene an AWS OpsWorks Stacks-Instances an. EBS-Volumes sind spezifisch für eine Availability Zone. Wenn Sie ein EBS-Volume in keinem Stack in einer bestimmten Region und Availability Zone verwenden, können Sie das Volume löschen. Weitere Informationen zum Löschen eines Amazon EBS-Volumes finden Sie unter Löschen eines Amazon EBS-Volumes im Amazon EC2 EC2-Benutzerhandbuch.

Buckets für Amazon Simple Storage Service (Amazon S3)

In AWS OpsWorks Stacks können Sie Amazon S3 S3-Buckets für Folgendes verwenden. Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

- Speichern von App-Code
- Speichern von Rezeptbüchern und Rezepten
- CloudTrail Protokolle, wenn Sie die CloudTrail Protokollierung in Stacks aktiviert haben AWS OpsWorks
- Amazon CloudWatch Logs-Streams, sofern Sie sie in AWS OpsWorks Stacks aktiviert haben

Elastic-IP-Adressen

Wenn Sie Elastic IP-Adressen bei AWS OpsWorks Stacks registriert haben und die Elastic IP-Adressen nicht mehr benötigen, können Sie die Elastic IP-Adresse freigeben.

Elastic Load Balancing-Load Balancer

Wenn Sie einen klassischen Elastic Load Balancing Load Balancer, den Sie mit Ebenen in Ihrem Stack verwendet haben, nicht mehr benötigen, können Sie ihn löschen. Weitere Informationen finden Sie unter Ihren Classic Load Balancer löschen im Benutzerhandbuch für Classic Load Balancer.

Amazon Relational Database Service (Amazon RDS) -Instances

Wenn Sie Amazon RDS-Datenbank-Instances (DB) bei AWS OpsWorks Stacks registriert haben und diese nicht mehr benötigen, können Sie DB-Instances löschen. Weitere Informationen zum Löschen von DB-Instances finden Sie unter Löschen einer DB-Instance im Amazon RDS-Benutzerhandbuch.

Amazon Elastic Container Service (Amazon ECS) -Cluster

Wenn Ihr Stack ECS-Cluster-Layer enthalten hat und Sie den bei einem Layer registrierten ECS-Cluster nicht mehr verwenden, können Sie den ECS-Cluster löschen. Weitere Informationen zum Löschen eines ECS-Clusters finden Sie unter Löschen eines Clusters im Amazon ECS Developer Guide.

Ebenen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Jeder Stack umfasst einen oder mehrere Layer, die jeweils eine Stack-Komponente repräsentieren, z. B. einen Load Balancer oder eine Gruppe von Anwendungsservern.

Beachten Sie bei der Arbeit mit AWS OpsWorks Stacks-Ebenen Folgendes:

- Jeder Layer in einem Stack muss mindestens eine Instance haben und kann optional jedoch auch mehrere Instances aufweisen.
- Jede Instance in einem Stack muss ein Element von mindestens einem Layer sein, mit Ausnahme von registrierten Instances.

  Sie können eine Instance nicht direkt konfigurieren, mit Ausnahme von einigen grundlegenden Einstellungen wie dem SSH-Schlüssel und dem Hostnamen. Sie müssen einen geeigneten Layer erstellen und konfigurieren und die Instance dem Layer hinzufügen.

Amazon EC2 EC2-Instances können optional Mitglied mehrerer Ebenen sein. In diesem Fall führt AWS OpsWorks Stacks die Rezepte zum Installieren und Konfigurieren von Paketen, Bereitstellen von Anwendungen usw. für jede Ebene der Instance aus.

Wenn Sie eine Instance mehreren Layern zuweisen, können Sie folgende Schritte durchführen:

- Senken Sie Kosten, indem Sie den Datenbankserver und den Load Balancer auf einer einzelnen Instance hosten.
- Verwenden Sie einen Ihrer Anwendungsserver für die Verwaltung.

  Erstellen Sie einen benutzerdefinierten Verwaltungs-Layer und fügen Sie eine der Anwendungsserver-Instances hinzu. Die Rezepte des Verwaltungs-Layers konfigurieren die Anwendungsserver-Instance, um Verwaltungsaufgaben durchzuführen und jede zusätzlich erforderliche Software zu installieren. Die anderen Anwendungsserver-Instances sind nur Anwendungsserver.

In diesem Abschnitt wird beschrieben, wie Sie mit Layern arbeiten.

Themen

OpsWorks Grundlagen der Ebene

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Abschnitt wird beschrieben, wie Operationen ausgeführt werden, die allen AWS OpsWorks Stacks-Ebenen gemeinsam sind.

Themen

- Eine OpsWorks Ebene erstellen
- Bearbeiten der Konfiguration einer Ebene OpsWorks
- Verwenden von Auto Healing zum Austausch fehlgeschlagener Instances
- Löschen einer Ebene OpsWorks

Eine OpsWorks Ebene erstellen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn Sie einen neuen Stack erstellen, wird folgende Seite angezeigt:

Um die erste Ebene hinzuzufügen OpsWorks

1. Klicken Sie auf Add a layer (Einen Layer hinzufügen).
2. Wählen Sie auf der Seite Add Layer (Einen Layer hinzufügen) den entsprechenden Layer aus, auf dem die Konfigurationsoptionen des Layers angezeigt werden.
3. Konfigurieren Sie den Layer entsprechend und klicken Sie auf Add Layer (Einen Layer hinzufügen), um ihn zum Stack hinzuzufügen. In den folgenden Abschnitten wird die Konfiguration der verschiedenen Layer beschrieben.

> **Note**
>
> Die Seite Add Layer (Einen Layer hinzufügen) zeigt nur die am häufigsten verwendeten Konfigurationseinstellungen für jeden Layer an. Sie können zusätzliche Einstellungen durch Bearbeiten des Layers festlegen.

4. Fügen Sie Instances zum Layer hinzu und starten Sie sie.

> **Note**
>
> Wenn eine Instance zu mehreren Layern gehört, müssen Sie sie zu allen Layern hinzufügen, bevor Sie die Instance starten. Sie können eine Online-Instance nicht zu einem Layer hinzufügen.

Um weitere Layer hinzuzufügen, öffnen Sie die Seite Layers (Layers) und klicken Sie auf + Layer (+ Layer), um die Seite Add Layer (Layer hinzufügen) zu öffnen.

Wenn Sie eine Instanz starten, führt AWS OpsWorks Stacks automatisch die Setup- und Deploy-Rezepte für jede Ebene der Instanz aus, um die entsprechenden Pakete zu installieren und zu konfigurieren und die entsprechenden Anwendungen bereitzustellen. Sie können den Einrichtungs- und Konfigurationsprozess einer Ebene auf verschiedene Weise anpassen, z. B. indem Sie den entsprechenden Lebenszyklusereignissen benutzerdefinierte Rezepte zuweisen. AWS OpsWorks Stacks führt benutzerdefinierte Rezepte nach den Standardrezepten für jedes Ereignis aus. Weitere Informationen finden Sie unter Cookbooks und Rezepte.

In den folgenden layerspezifischen Abschnitten wird beschrieben, wie die Schritte 2 und 3 für die verschiedenen AWS OpsWorks Stacks-Ebenen gehandhabt werden. Weitere Informationen zum Hinzufügen von Instances finden Sie unter Hinzufügen einer Instance zu einem Layer.

Bearbeiten der Konfiguration einer Ebene OpsWorks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Nach dem Erstellen eines Layers sind einige Eigenschaften (z. B. AWS-Region) unveränderbar, aber Sie können die meisten Parameter der Layer-Konfiguration jederzeit ändern. Beim Bearbeiten des Layers besteht auch Zugriff auf Konfigurationseinstellungen, die nicht auf der Seite Add Layer (Layer hinzufügen) verfügbar sind. Die Einstellungen werden wirksam, sobald Sie die neue Konfiguration speichern.

Um eine Ebene zu bearbeiten OpsWorks

1. Klicken Sie im Navigationsbereich auf Layers (Layers).
2. Wählen Sie auf der Seite Layers (Layers) den Namen eines Layers aus, um die Detailseite mit der aktuellen Konfiguration zu öffnen.

   > **Note**
   >
   > Durch Auswahl eines Namens unterhalb des Layer-Namens werden Sie direkt zur entsprechenden Registerkarte auf der Detailseite geleitet.

3. Klicken Sie auf Edit (Bearbeiten) und wählen Sie dann die entsprechende Registerkarte aus: General Settings (Allgemeine Einstellungen), Recipes (Rezepte), Network (Netzwerk), EBS Volumes (EBS-Volumes) oder Security (Sicherheit).

In den folgenden Abschnitten werden die Einstellungen für verschiedene, in allen Layern verfügbare Registerkarten beschrieben. Einige Layer haben zusätzliche, Layer-spezifische Einstellungen, die oben auf der Seite angezeigt werden. Darüber hinaus sind einige Einstellungen nur für Linux-basierte Stacks verfügbar, wie angegeben.

Themen

- CloudWatch Logs
- Tags

Allgemeine Einstellungen

Alle Layer haben die folgenden Einstellungen:

Auto Healing aktiviert

Bestimmt, ob Auto Healing (Auto Healing) für die Instances des Layers aktiviert ist. Die Standardeinstellung ist Yes (Ja).

Custom JSON

Daten im JSON-Format, die an die Chef-Rezepte für alle Instances in diesem Layer übergeben werden. Auf diese Weise können Sie beispielsweise Daten an Ihre eigenen Rezepte übergeben. Weitere Informationen finden Sie unter Nutzen eines benutzerdefinierten JSON-Objekts.

> **Note**
>
> Sie können benutzerdefinierte JSON-Objekte auf Bereitstellungs-, Layer- und Stacks-Ebene deklarieren. Diese Lösung bietet sich an, wenn Sie möchten, dass einige benutzerdefinierte JSON-Objekte im gesamten Stack oder nur für eine einzelne Bereitstellung sichtbar sind. Oder Sie möchten die auf Layer-Ebene deklarierte benutzerdefinierte JSON-Objekte mit einem benutzerdefinierten JSON-Objekt auf Bereitstellungsebene überschreiben. Wenn Sie benutzerdefinierte JSON-Objekte auf mehreren Ebenen deklarieren, werden die auf Layer- und Stack-Ebene deklarierten benutzerdefinierten JSON-Objekte von dem auf der Bereitstellungsebene deklarierten benutzerdefinierten JSON-Objekt überschrieben. Benutzerdefiniertes JSON-Objekt, das nur auf Stack-Ebene deklariert ist, wird von jeglichen, nur auf Layer-Ebene deklarierten benutzerdefinierten JSON-Objekten überschrieben.
> Um mit der AWS OpsWorks Stacks-Konsole benutzerdefiniertes JSON für eine Bereitstellung anzugeben, wählen Sie auf der Seite App bereitstellen die Option Erweitert aus. Geben Sie das benutzerdefinierte JSON-Objekt in das Feld Custom Chef JSON (Benutzerdefiniertes Chef JSON) ein und wählen Sie dann Save (Speichern) aus.
> Um mit der AWS OpsWorks Stacks-Konsole benutzerdefiniertes JSON für einen Stack anzugeben, geben Sie auf der Seite mit den Stack-Einstellungen das benutzerdefinierte JSON in das Feld Benutzerdefiniertes JSON ein und wählen Sie dann Speichern aus.

Weitere Informationen finden Sie unter Nutzen eines benutzerdefinierten JSON-Objekts und Bereitstellen von Anwendungen.

Zeitbeschränkung beim Instance-Shutdown

Gibt an, wie lange (in Sekunden) AWS OpsWorks Stacks wartet, nachdem es ein Shutdown-Lifecycle-Ereignis ausgelöst hat, bevor es die Amazon EC2 EC2-Instance stoppt oder beendet. Der Standardwert beträgt 120 Sekunden. Mit dieser Einstellung erhalten die Shutdown-Rezepte der Instance ausreichend Zeit, ihre Aufgaben abzuschließen, bevor die Instance beendet wird. Wenn Ihre benutzerdefinierten Shutdown-Rezepte mehr Zeit benötigen, ändern Sie die Einstellung entsprechend. Weitere Informationen zum Instance-Shutdown finden Sie unter Anhalten einer Instance.

Die restlichen Einstellungen auf dieser Registerkarte variieren je nach Layer-Typ und sind mit den Einstellungen auf der Seite Add Layer (Layer hinzufügen) des Layers identisch.

Rezepte

Die Registerkarte Recipes (Rezepte) enthält folgende Einstellungen.

Custom Chef recipes (Benutzerdefinierte Chef-Rezepte)

Sie können den Lebenszyklusereignissen eines Layers benutzerdefinierte Chef-Rezepte zuweisen. Weitere Informationen finden Sie unter Ausführen von Rezepten.

Network (Netzwerk)

Die Registerkarte Network (Netzwerk) enthält folgende Einstellungen.

Elastic Load Balancing

Sie können einen Elastic Load Balancing Load Balancer einem beliebigen Layer anfügen. AWS OpsWorks Stacks registriert dann automatisch die Online-Instances des Layers beim Load Balancer und deregistriert sie, wenn sie offline gehen. Wenn Sie die Funktion zum Entleeren von Verbindungen des Load Balancers aktiviert haben, können Sie angeben, ob Stacks sie unterstützt. AWS OpsWorks Weitere Informationen finden Sie unter Elastic Load Balancing Lastenausgleichsebene.

Automatically Assign IP Addresses (IP-Adressen automatisch zuweisen)

Sie können steuern, ob AWS OpsWorks Stacks den Instances des Layers automatisch öffentliche oder elastische IP-Adressen zuweist. Folgendes geschieht, wenn Sie diese Option aktivieren:

- AWS OpsWorks Stacks weist beispielsweise bei jedem Start der Instance automatisch eine Adresse zu.
- Für Amazon EBS-gestützte Instances weist AWS OpsWorks Stacks automatisch eine Adresse zu, wenn die Instance zum ersten Mal gestartet wird.
- Wenn eine Instance zu mehr als einer Ebene gehört, weist AWS OpsWorks Stacks automatisch eine Adresse zu, wenn Sie die automatische Zuweisung für mindestens eine der Ebenen aktiviert haben.

> **Note**
>
> Wenn Sie die automatische Zuweisung von öffentlichen IP-Adressen aktivieren, gilt dies nur für neue Instanzen. AWS OpsWorks Stacks können die öffentliche IP-Adresse für bestehende Instanzen nicht aktualisieren.

Wenn Ihr Stack in einer VPC ausgeführt wird, verfügen Sie über separate Einstellungen für öffentliche und Elastic IP-Adressen. In der folgenden Tabelle wird beschrieben, wie diese interagieren:

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**
>
> Instanzen müssen über eine Möglichkeit verfügen, mit dem AWS OpsWorks Stacks-Dienst, den Linux-Paket-Repositorys und den Cookbook-Repositorys zu kommunizieren. Wenn Sie keine öffentlichen IP-Adressen oder Elastic IP-Adresse angeben, muss VPC ein Element wie z. B. NAT enthalten, damit die Instance des Layers mit externen Stellen

kommunizieren kann. Weitere Informationen finden Sie unter Ausführen eines Stacks in einer VPC.

Wenn der Stack nicht in einem VPC ausgeführt wird, ist Elastic IP addresses (Elastic IP-Adressen) die einzige verfügbare Einstellungsoption:

- Yes (Ja): Instances erhalten eine Elastic IP-Adresse, wenn sie zum ersten Mal gestartet werden, oder eine öffentliche IP-Adresse, wenn die Zuweisung einer Elastic IP-Adresse nicht möglich ist.
- No (Nein): Instances erhalten bei jedem Start eine öffentliche IP-Adresse.

EBS-Datenträger

Die Registerkarte EBS Volumes (EBS-Volumes) enthält folgende Einstellungen.

EBS-optimierte Instances

Ob die Instances des Layers für Amazon Elastic Block Store (Amazon EBS) optimiert werden sollen. Weitere Informationen finden Sie unter Amazon EBS-Optimierte Instances.

Additional EBS Volumes (Zusätzliche EBS-Volumes)

(Nur Linux) Sie können Amazon EBS-Volumes zu den Instances des Layers hinzufügen oder aus ihnen entfernen. Wenn Sie eine Instance starten, erstellt AWS OpsWorks Stacks automatisch die Volumes und hängt sie den Instances an. Sie können auf der Seite Resources (Ressourcen) die EBS-Volumes eines Stacks verwalten. Weitere Informationen finden Sie unter Ressourcenmanagement.

- Bereitstellungspunkt — (Erforderlich) Geben Sie den Bereitstellungspunkt oder das Verzeichnis an, in dem das EBS-Volume bereitgestellt werden soll.
- # Festplatten — (Optional) Wenn Sie ein RAID-Array angegeben haben, die Anzahl der Festplatten im Array.

  Jeder RAID-Layer verfügt über eine standardmäßige Anzahl von Festplatten. Sie können jedoch eine größere Anzahl aus der Liste auswählen.
- Gesamtgröße (GiB) — (Erforderlich) Die Größe des Volumes in GiB.

  Für ein RAID-Array gibt diese Einstellung die gesamte Array-Größe und nicht die Größe der einzelnen Festplatten an.

Die folgende Tabelle zeigt die Mindest- und die maximale Volume-Größe, die für die einzelnen Volume-Typen zulässig sind.

Volume-Typ	Mindestgröße (GiB)	Maximale Größe (GiB)
Magnetic	1	1024
Bereitgestellte IOPS (SSD)	4	16384
Allzweck (SSD)	1	16384
Throughput Optimized (HDD)	500	16384
Cold HDD	500	16384

- Datenträgertyp — (Optional) Geben Sie an, ob ein magnetisches Allzweck-SSD-Volume, ein durchsatzoptimiertes HDD-, Cold HDD- oder PIOPS-Volume erstellt werden soll.

  Der Standardwert ist Magnetic (Magnetisch).
- Verschlüsselt — (Optional) Geben Sie an, ob der Inhalt des EBS-Volumes verschlüsselt werden soll.
- IOPS pro Festplatte — (Erforderlich für bereitgestellte IOPS-SSD- und Allzweck-SSD-Volumes) Wenn Sie ein bereitgestelltes IOPS-SSD- oder Allzweck-SSD-Volume angeben, müssen Sie auch die IOPS pro Festplatte angeben.

  Bei bereitgestellten IOPS-Volumes können Sie beim Erstellen des Volumes die IOPS-Rate angeben. Das Verhältnis zwischen der bereitgestellten IOPS und der Volume-Größe darf maximal 30 betragen (d. h., ein Volume mit 3 000 IOPS muss mindestens 100 GB groß sein). Für Allzweck-Volumes-Typen (SSD) beträgt die IOPS-Basisleistung die dreifache Volume-Größe mit maximal 10 000 IOPS, die über einen Zeitraum von 30 Minuten um 3 000 IOPS erweitert werden kann.

Wenn Sie Volumes aus einem Layer hinzufügen oder entfernen möchten, beachten Sie Folgendes:

- Wenn Sie ein Volume hinzufügen, erhält jede neue Instance Zugriff auf das neue Volume, allerdings aktualisiert AWS OpsWorks Stacks nicht die vorhandenen Instances.

- Wenn Sie ein Volume entfernen, gilt dies nur für neue Instances. Die vorhandenen Instances behalten ihre Volumes.

Angabe eines Mounting-Punkts

Sie können einen beliebigen Mounting-Punkt angeben. Beachten Sie jedoch, dass einige Bereitstellungspunkte für die Verwendung durch AWS OpsWorks Stacks oder Amazon EC2 reserviert sind und nicht für Amazon EBS-Volumes verwendet werden sollten. Verwenden Sie keine typischen Linux-Systemordner wie `/home` oder `/etc`.

Die folgenden Bereitstellungspunkte sind für die Verwendung durch Stacks reserviert. AWS OpsWorks

- `/srv/www`
- `/var/log/apache2` (Ubuntu)
- `/var/log/httpd` (Amazon Linux)
- `/var/log/mysql`
- `/var/www`

Beim Start oder Neustart einer Instance verwendet autofs (Daemon für automatisches Mounting) flüchtige Geräte-Mounting-Punkte wie z. B. `/media/ephemeral0` zum Erstellen von Verzeichnissen (bind mounts). Dieser Vorgang findet statt, bevor Amazon EBS-Volumes bereitgestellt werden. Um sicherzustellen, dass der Bereitstellungspunkt Ihres Amazon EBS-Volumes nicht mit autofs kollidiert, geben Sie keinen temporären Geräte-Einhängepunkt an. Die möglichen Einhängepunkte für kurzlebige Geräte hängen vom jeweiligen Instance-Typ ab und davon, ob es sich um einen Instance-Store oder einen Amazon EBS-gestützten Instance-Speicher handelt. Um Konflikte mit autofs zu vermeiden, gehen Sie folgendermaßen vor:

- Überprüfen Sie die flüchtigen Geräte-Mounting-Punkte für den bestimmten Instance-Typ und den zu verwendenden Sicherungsspeicher.
- Beachten Sie, dass ein Bereitstellungspunkt, der für eine vom Instance-Speicher unterstützte Instance funktioniert, zu Konflikten mit autofs führen kann, wenn Sie zu einer Amazon EBS-gestützten Instance wechseln oder umgekehrt.

> **Note**
>
> Wenn Sie die Instance-Speicher-Blockgerät-Zuweisung ändern möchten, können Sie ein benutzerdefiniertes AMI erstellen. Weitere Informationen finden Sie unter Amazon EC2-Instance-Speicher. Weitere Informationen zum Erstellen eines benutzerdefinierten AMI für AWS OpsWorks Stacks finden Sie unterVerwenden von benutzerdefinierten AMIs.

Das folgende Beispiel zeigt, wie Sie mit einem benutzerdefinierten Rezept sicherstellen, dass ein Volume-Mounting-Punkt nicht mit autofs in Konflikt gerät. Sie können es für Ihren speziellen Anwendungsfall anpassen.

So vermeiden Sie einen konfliktiven Mounting-Punkt

1. Weisen Sie der gewünschten Ebene ein Amazon EBS-Volume zu, verwenden Sie jedoch einen Bereitstellungspunkt`/mnt/workspace`, der niemals zu Konflikten mit autofs führt.
2. Implementieren Sie das folgende benutzerdefinierte Rezept, das ein Anwendungsverzeichnis auf dem Amazon EBS-Volume erstellt und von dort aus `/srv/www/` Links zu diesem enthält. Weitere Informationen zur Implementierung von benutzerdefinierten Rezepten finden Sie unter Cookbooks und Rezepte und Stacks anpassen 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. Fügen Sie eine `depends 'deploy'`-Zeile in die Datei des benutzerdefinierten Rezeptbuchs `metadata.rb` hinzu.
4. Weisen Sie dieses Rezept dem Einrichtungsereignis des Layers zu.

Sicherheit

Die Registerkarte Security (Sicherheit) enthält folgende Einstellungen.

Sicherheitsgruppen

Einem Layer muss mindestens eine Sicherheitsgruppe zugewiesen sein. Sie geben an, wie Sicherheitsgruppen verknüpft werden sollen, wenn Sie einen Stack erstellen oder aktualisieren. AWS OpsWorks Stacks bietet einen Standardsatz integrierter Sicherheitsgruppen.

- Die Standardoption besteht darin, dass AWS OpsWorks Stacks jeder Ebene automatisch die entsprechende integrierte Sicherheitsgruppe zuordnet.
- Sie können auch die nicht automatische Zuweisung von Sicherheitsgruppen auswählen und stattdessen jedem Layer bei dessen Erstellung eine benutzerdefinierte Sicherheitsgruppe zuweisen.

Weitere Informationen zu Sicherheitsgruppen finden Sie unter Verwenden von Sicherheitsgruppen.

Nachdem der Layer erstellt wurde, können Sie mit Security Groups (Sicherheitsgruppen) mehrere Sicherheitsgruppen zum Layer hinzufügen, indem Sie sie aus der Liste Custom security groups (Benutzerdefinierte Sicherheitsgruppen) auswählen. Nachdem Sie einer Ebene eine Sicherheitsgruppe hinzugefügt haben, fügt AWS OpsWorks Stacks sie allen neuen Instanzen hinzu. (Beachten Sie, dass Instance-Store-Instances, die neu gestartet werden, als neue Instanzen angezeigt werden, sodass sie auch über die neuen Sicherheitsgruppen verfügen.) AWS OpsWorks Stacks fügt Online-Instances keine Sicherheitsgruppen hinzu.

Sie können vorhandene Sicherheitsgruppen löschen, indem Sie auf x klicken. Gehen Sie dazu folgendermaßen vor:

- Wenn Sie festlegen, dass AWS OpsWorks Stacks integrierte Sicherheitsgruppen automatisch zuordnen soll, können Sie benutzerdefinierte Sicherheitsgruppen, die Sie zuvor hinzugefügt haben, löschen, indem Sie auf das X klicken, aber Sie können die integrierte Gruppe nicht löschen.

- Wenn Sie sich dazu entscheiden, die integrierten Sicherheitsgruppen nicht automatisch zuzuweisen, können Sie alle existierenden Sicherheitsgruppen, auch die ursprünglichen, löschen, solange dem Layer mindesten eine Gruppe zugeordnet ist.

Nachdem Sie eine Sicherheitsgruppe aus einer Ebene entfernt haben, fügt AWS OpsWorks Stacks sie keinen neuen oder neu gestarteten Instanzen hinzu. AWS OpsWorks Stacks entfernt keine Sicherheitsgruppen aus Online-Instanzen.

> **Note**
>
> Wenn Ihr Stack in einer VPC ausgeführt wird, können Sie mithilfe der Amazon EC2 EC2-Konsole, API oder CLI eine Sicherheitsgruppe für eine Online-Instance hinzufügen oder entfernen. Diese Sicherheitsgruppe wird jedoch in der AWS OpsWorks Stacks-Konsole nicht sichtbar sein. Wenn Sie die Sicherheitsgruppe entfernen möchten, müssen Sie auch Amazon EC2 verwenden. Weitere Informationen finden Sie unter Sicherheitsgruppen.

Beachten Sie Folgendes:

- Sie können die Zugriffseinstellungen des Ports der integrierten Sicherheitsgruppe durch Hinzufügen einer restriktiveren Sicherheitsgruppe nicht einschränken. Wenn es mehrere Sicherheitsgruppen gibt, verwendet Amazon EC2 die tolerantesten Einstellungen.
- Ändern Sie nicht die Konfiguration einer integrierten Sicherheitsgruppe. Wenn Sie einen Stack erstellen, überschreibt AWS OpsWorks Stacks die Konfigurationen der integrierten Sicherheitsgruppen, sodass alle Änderungen, die Sie vornehmen, verloren gehen, wenn Sie das nächste Mal einen Stack erstellen.

Wenn Sie feststellen, dass Sie restriktivere Sicherheitsgruppen-Einstellungen für einen oder mehrere Layer benötigen, führen Sie die folgenden Schritte aus:

1. Erstellen Sie benutzerdefinierte Sicherheitsgruppen mit den entsprechenden Einstellungen und fügen Sie sie zu den entsprechenden Layern hinzu.

   Jeder Layer in Ihrem Stack muss mindestens über eine Sicherheitsgruppe zusätzlich zu der integrierten Gruppe verfügen, auch wenn nur für einen Layer benutzerdefinierte Einstellungen erforderlich sind.

2. Bearbeiten Sie die Stack-Konfiguration und setzen Sie die Einstellung OpsWorksSicherheitsgruppen verwenden auf Nein.

AWS OpsWorks Stacks entfernt automatisch die integrierte Sicherheitsgruppe aus jeder Ebene.

Weitere Informationen über Sicherheitsgruppen finden Sie unter Amazon EC2-Sicherheitsgruppen.

EC2 Instance Profile (EC2-Instance-Profile)

Sie können das EC2-Profil für Instances des Layers ändern. Weitere Informationen finden Sie unter Festlegen von Berechtigungen für Apps auf EC2-Instances.

CloudWatch Logs

Auf der Registerkarte CloudWatch Logs können Sie Amazon CloudWatch Logs aktivieren oder deaktivieren. CloudWatch Die Logs-Integration funktioniert mit den Linux-basierten Stacks Chef 11.10 und Chef 12. Weitere Informationen zur Aktivierung der CloudWatch Log-Integration und zur Angabe der Logs, die Sie in der CloudWatch Logs-Konsole verwalten möchten, finden Sie unter. Amazon CloudWatch Logs mit AWS OpsWorks Stacks verwenden

Tags

Mit der Registerkarte Tags können Sie Kostenzuordnungs-Tags für Ihren Layer anwenden. Nachdem Sie Tags hinzugefügt haben, können Sie sie in der AWS Billing and Cost Management Konsole aktivieren. Wenn Sie ein Tag erstellen, wenden Sie das Tag auf alle Ressourcen innerhalb der gekennzeichneten Struktur an. Wenn Sie beispielsweise ein Tag auf eine Ebene anwenden, wenden Sie das Tag auf jede Instance, jedes Amazon EBS-Volume oder jeden Elastic Load Balancing Load Balancer in der Ebene an. Weitere Informationen dazu, wie Sie Ihre Tags aktivieren und damit die Kosten Ihrer AWS OpsWorks Stacks-Ressourcen verfolgen und verwalten können, finden Sie unter Verwenden von Kostenzuordnungs-Tags und Aktivieren von benutzerdefinierten Kostenzuordnungs-Tags im Billing and Cost Management-Benutzerhandbuch. Weitere Informationen zum Erstellen von Tags in AWS OpsWorks Stacks finden Sie unter Tags.

Verwenden von Auto Healing zum Austausch fehlgeschlagener Instances

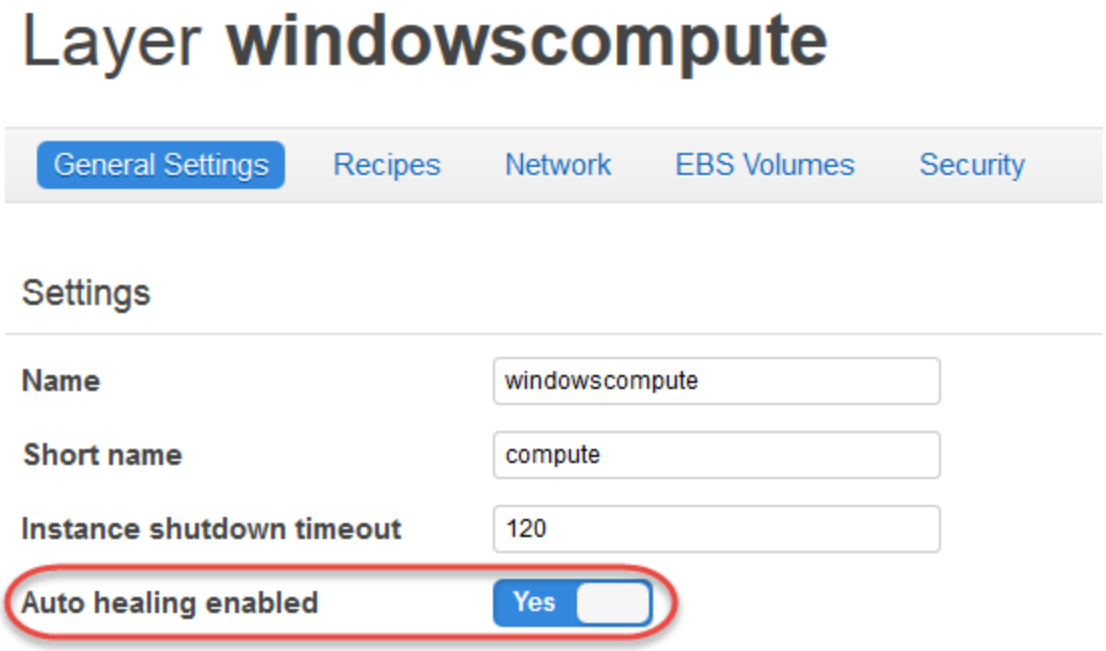
Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren.

Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Jede Instanz hat einen AWS OpsWorks Stacks-Agenten, der regelmäßig mit dem Service kommuniziert. AWS OpsWorks Stacks verwendet diese Kommunikation, um den Zustand der Instance zu überwachen. Wenn ein Agent länger als etwa fünf Minuten nicht mit dem Service kommuniziert, geht AWS OpsWorks Stacks davon aus, dass die Instanz ausgefallen ist.

Auto Healing wird auf Layer-Ebene festgelegt. Sie können Auto Healing auch festlegen, indem Sie die Einstellungen des Layers bearbeiten, wie in der nachstehenden Abbildung dargestellt.



Note

Eine Instance kann zu mehreren Layern gehören. Wenn bei einer dieser Ebenen die auto Heilung deaktiviert ist, repariert AWS OpsWorks Stacks die Instanz nicht, wenn sie fehlschlägt.

Wenn für eine Ebene die auto Heilung aktiviert ist (Standardeinstellung), ersetzt AWS OpsWorks Stacks die ausgefallenen Instanzen der Ebene automatisch wie folgt:

Instance-Speicher-gestützte Instance

1. Stoppt die Amazon EC2 EC2-Instance und überprüft, ob sie heruntergefahren wurde.

2. Löscht die Daten auf dem Stamm-Volume.
3. Erstellt eine neue Amazon EC2 EC2-Instance mit demselben Hostnamen, derselben Konfiguration und derselben Layer-Mitgliedschaft.
4. Fügt alle Amazon EBS-Volumes erneut an, einschließlich Volumes, die nach dem ursprünglichen Start der alten Instance angehängt wurden.
5. Weist eine neue öffentliche und private IP-Adresse zu.
6. Wenn die alte Instance mit einer Elastic IP-Adresse verknüpft war, wird die neue Instance mit derselben IP-Adresse verknüpft.

Amazon EBS-gestützte Instance

1. Stoppt die Amazon EC2 EC2-Instance und überprüft, ob sie gestoppt wurde.
2. Startet die EC2-Instance.

Nachdem die automatisch reparierte Instanz wieder online ist, löst AWS OpsWorks Stacks ein Configure Lifecycle-Ereignis für alle Instanzen des Stacks aus. Die zugeordneten Stack-Konfigurations- und Bereitstellungsattribute enthalten die öffentlichen und privaten IP-Adressen der Instance. Mit benutzerdefinierten Konfigurationsrezepten können neue IP-Adressen vom Knotenobjekt bezogen werden.

Wenn Sie ein Amazon EBS-Volume für die Instances einer Ebene angeben, erstellt AWS OpsWorks Stacks ein neues Volume und hängt es jeder Instance an, wenn die Instance gestartet wird. Wenn Sie das Volume später von einer Instance trennen möchten, verwenden Sie die Seite Resources (Ressourcen).

Wenn AWS OpsWorks Stacks eine Instanz einer Ebene auto heilt, werden Volumen wie folgt behandelt:

- Wenn das Volume an die Instanz angehängt wurde, als die Instanz ausfiel, werden das Volume und seine Daten gespeichert, und AWS OpsWorks Stacks fügt es der neuen Instanz hinzu.
- Wenn das Volume zum Zeitpunkt des Fehlschlagens der Instance dieser nicht zugewiesen war, erstellt AWS OpsWorks Stacks ein neues, leeres Volume mit der von dem Layer definierten Konfiguration und ordnet es der neuen Instance zu.

Auto Healing ist standardmäßig aktiviert, aber Sie können es durch Bearbeiten der allgemeinen Einstellungen des Layers deaktivieren.

> **Important**
>
> Wenn Sie Auto Healing aktiviert haben, stellen Sie sicher, dass Sie die folgenden Schritte ausführen:
>
> - Verwenden Sie nur die AWS OpsWorks Stacks-Konsole, CLI oder API, um Instanzen zu stoppen.
>
> Wenn Sie eine Instance auf andere Weise beenden, z. B. über die Amazon EC2 EC2-Konsole, behandelt AWS OpsWorks Stacks die Instance als ausgefallen und repariert sie auto.
> - Verwenden Sie Amazon EBS-Volumes, um alle Daten zu speichern, die Sie nicht verlieren möchten, wenn die Instance auto repariert wird.
>
> Auto Healing stoppt die alte Amazon EC2 EC2-Instance, wodurch alle Daten zerstört werden, die nicht auf einem Amazon EBS-Volume gespeichert sind. Amazon EBS-Volumes werden wieder an die neue Instance angehängt, wodurch alle gespeicherten Daten erhalten bleiben.

Löschen einer Ebene OpsWorks

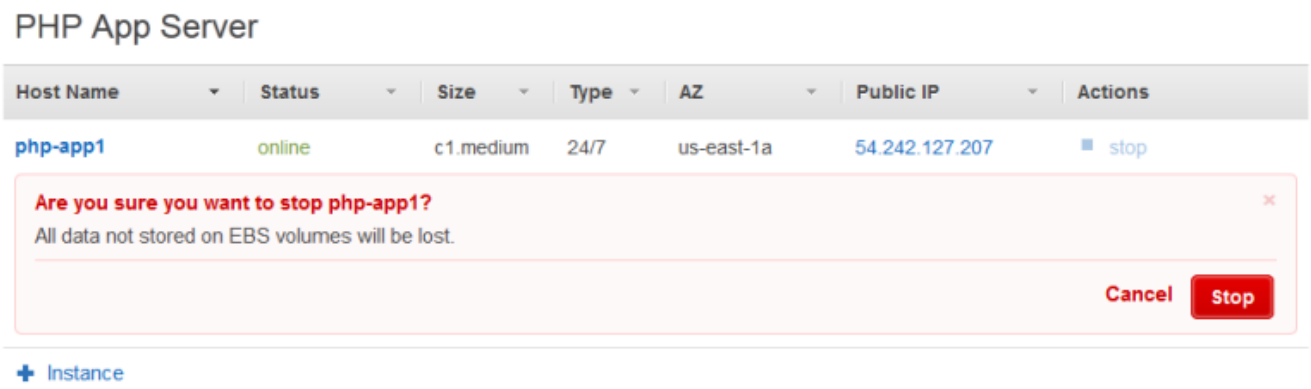
> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn du eine AWS OpsWorks Stacks-Ebene nicht mehr benötigst, kannst du sie aus deinem Stack löschen.

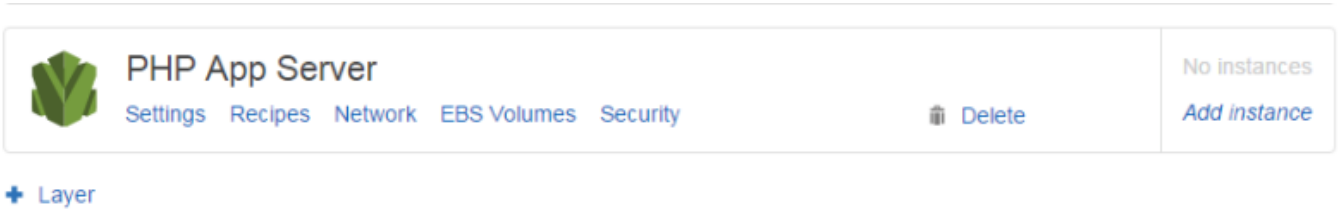
Um eine OpsWorks Ebene zu löschen

1. Klicken Sie im Navigationsbereich auf Instances.

2. Klicken Sie auf der Seite Instances unterhalb des Namens des zu löschenden Layers für jede Instance in der Spalte Actions auf stop.



3. Nachdem jede Instance beendet worden ist, klicken Sie auf delete (Löschen), um es aus dem Layer zu entfernen.
4. Klicken Sie im Navigationsbereich auf Layers (Layers).
5. Wählen Sie auf der Seite Layers (Layers) die Option Delete (Löschen) aus.



Elastic Load Balancing Lastenausgleichsebene

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Elastic Load Balancing funktioniert etwas anders als eine AWS OpsWorks Stacks-Ebene. Anstatt eine Ebene zu erstellen und ihr Instances hinzuzufügen, verwenden Sie die Elastic Load Balancing Balancing-Konsole oder API, um einen Load Balancer zu erstellen und ihn dann an

einen vorhandenen Layer anzuhängen. Elastic Load Balancing verteilt nicht nur den Traffic auf die Instances des Layers, sondern macht auch Folgendes:

- Erkennt fehlerhafte Amazon EC2 EC2-Instances und leitet den Datenverkehr an die verbleibenden fehlerfreien Instances weiter, bis die fehlerhaften Instances wiederhergestellt wurden.
- Er skaliert als Reaktion auf den eingehenden Datenverkehr automatisch die Kapazität zur Anforderungsbearbeitung.
- Wenn Sie die Funktion Connection Draining aktivieren, schickt der Load Balancer keine neuen Anforderungen mehr an fehlerhafte oder abgemeldete Instances, sondern hält die Verbindung bis zu einem festgelegten Zeitüberschreitungswert aufrecht, damit die Instances laufende Anforderungen abschließen können.

Nachdem Sie einen Load Balancer an eine Ebene angehängt haben, geht Stacks wie folgt vor: AWS OpsWorks

- Meldet alle derzeit registrierten Instances ab.
- Registriert automatisch die Instances der Ebene, einschließlich last- und zeitbasierter Instances, wenn sie online gehen, und meldet sie automatisch wieder ab, wenn sie offline gehen.
- Startet automatisch das Routing von Anfragen an registrierte Instances in ihren Availability Zones.

Wenn Sie die Funktion zum Entleeren von Verbindungen des Load Balancers aktiviert haben, können Sie angeben, ob AWS OpsWorks Stacks sie unterstützt. Wenn Sie die Unterstützung für Verbindungsabbau aktivieren (Standardeinstellung), führt AWS OpsWorks Stacks nach dem Herunterfahren einer Instanz Folgendes aus:

- Meldet die Instance vom Load Balancer ab.

  Der Load Balancer sendet keine neuen Anforderungen mehr und startet den Verbindungsausgleich.
- Verzögert das Auslösen eines Shutdown-Lebenszyklusereignisses bis der Load Balancer den Verbindungsausgleich abgeschlossen hat.

Wenn Sie die Unterstützung für Verbindungsverlust nicht aktivieren, löst AWS OpsWorks Stacks das Shutdown-Ereignis aus, sobald die Instance heruntergefahren wird, auch wenn die Instance immer noch mit dem Load Balancer verbunden ist.

Um Elastic Load Balancing mit einem Stack zu verwenden, müssen Sie zunächst mithilfe der Elastic Load Balancing-Konsole, CLI oder API einen oder mehrere Load Balancer in derselben Region erstellen. Dabei sollten Sie Folgendes beachten:

- Sie können einem Layer nur einen Load Balancer anfügen.
- Jeder Load Balancer ist nur für einen Layer zuständig.
- AWS OpsWorks Stacks unterstützt den Application Load Balancer nicht. Sie können Classic Load Balancer nur mit AWS OpsWorks Stacks verwenden.

Das bedeutet, dass Sie für jede Ebene in jedem Stack, den Sie ausgleichen möchten, einen separaten Elastic Load Balancing Load Balancer erstellen und ihn nur für diesen Zweck verwenden müssen. Es wird empfohlen, jedem Elastic Load Balancing Load Balancer, den Sie mit AWS OpsWorks Stacks verwenden möchten, einen eindeutigen Namen zuzuweisen, z. B. MyStack 1-RailsLayer -ELB, um zu vermeiden, dass ein Load Balancer für mehr als einen Zweck verwendet wird.

> **Important**
>
> Wir empfehlen, neue Elastic Load Balancing Load Balancer für Ihre AWS OpsWorks Stacks-Layer zu erstellen. Wenn Sie sich dafür entscheiden, einen vorhandenen Elastic Load Balancing Load Balancer zu verwenden, sollten Sie zunächst sicherstellen, dass er nicht für andere Zwecke verwendet wird und keine angehängten Instances hat. Nachdem der Load Balancer an die Ebene angehängt wurde, werden alle vorhandenen Instances OpsWorks entfernt und der Load Balancer so konfiguriert, dass er nur die Instances der Ebene verarbeitet. Es ist zwar technisch möglich, die Elastic Load Balancing Balancing-Konsole oder API zu verwenden, um die Konfiguration eines Load Balancers zu ändern, nachdem er an eine Ebene angehängt wurde, aber Sie sollten dies nicht tun, da die Änderungen nicht dauerhaft sind.

So fügen Sie einem Layer einen Elastic Load Balancing Load Balancer hinzu

1. Falls Sie dies noch nicht getan haben, verwenden Sie die Elastic Load Balancing Balancing-Konsole, API oder CLI, um einen Load Balancer in der Region des Stacks zu erstellen. Wenn Sie einen Load Balancer erstellen, führen Sie die folgenden Schritte aus:

- Stellen Sie sicher, dass Sie eine Zustandsprüfung für den Ping-Pfad angeben, der für Ihre Anwendung geeignet ist.

  Das Standard-Ping-Pfad ist `/index.html`. Wenn Ihre Anwendung `index.html` nicht umfasst, müssen Sie einen entsprechenden Ping-Pfad angeben oder die Zustandsprüfung schlägt fehl.
- Wenn Sie die Funktion Connection Draining verwenden möchten, stellen Sie sicher, dass die Funktion aktiviert ist und einen geeigneten Zeitüberschreitungswert hat.

Weitere Informationen finden Sie unter Elastic Load Balancing.

2. Erstellen Sie den Layer, für den ein Verbindungausgleich erfolgen soll oder bearbeiten Sie die Netzwerk-Einstellungen eines vorhandenen Layers.

   > **Note**
   >
   > Sie können keinen Load Balancer anfügen, wenn Sie einen benutzerspezifischen Layer erstellen. Sie müssen die Layer-Einstellungen bearbeiten.

3. Wählen Sie unter Elastic Load Balancing den Load Balancer aus, den Sie an den Layer anhängen möchten, und geben Sie an, ob AWS OpsWorks Stacks Connection Draining unterstützen sollen.

Nachdem Sie einen Load Balancer an eine Ebene angehängt haben, löst AWS OpsWorks Stacks ein Configure-Lifecycle-Ereignis auf den Instances des Stacks aus, um sie über die Änderung zu informieren. AWS OpsWorks Stacks löst auch ein Configure-Ereignis aus, wenn Sie einen Load Balancer trennen.

> **Note**
>
> Nach dem Booten einer Instanz führt AWS OpsWorks Stacks die Setup- und Deploy-Rezepte aus, mit denen Pakete installiert und Anwendungen bereitgestellt werden. Nachdem diese Rezepte abgeschlossen sind, befindet sich die Instance im Online-Status und AWS OpsWorks Stacks registriert die Instance bei Elastic Load Balancing. AWS OpsWorks Stacks löst auch ein Configure-Ereignis aus, nachdem die Instance online gegangen ist. Das bedeutet, dass die Elastic Load Balancing Balancing-Registrierung und die Configure-Rezepte gleichzeitig ausgeführt werden können und die Instance möglicherweise registriert

wird, bevor die Configure-Rezepte abgeschlossen sind. Um sicherzustellen, dass ein Rezept abgeschlossen ist, bevor eine Instance bei Elastic Load Balancing registriert wird, sollten Sie das Rezept zu den Lifecycle-Ereignissen Setup oder Deploy des Layers hinzufügen. Weitere Informationen finden Sie unter Ausführen von Rezepten.

Manchmal ist es sinnvoll, eine Instance von einem Load Balancer zu entfernen. Wenn Sie beispielsweise eine Anwendung aktualisieren, empfehlen wir, dass Sie die Anwendung zunächst für eine einzelne Instance bereitstellen und prüfen, ob sie ordnungsgemäß funktioniert, bevor Sie die Anwendung für alle Instances bereitstellen. In der Regel entfernen Sie die Instance aus dem Load Balancer, sodass sie keine Benutzeranforderungen erhält, bis die Aktualisierung überprüft wurde.

Sie müssen die Elastic Load Balancing Balancing-Konsole oder API verwenden, um eine Online-Instance vorübergehend von einem Load Balancer zu entfernen. Im Folgenden wird beschrieben, wie Sie die Konsole verwenden.

Vorübergehendes Entfernen einer Instance von einem Load Balancer

1. Öffnen Sie die Amazon EC2 EC2-Konsole und wählen Sie Load Balancers.
2. Wählen Sie einen geeigneten Load Balancer und öffnen Sie die Registerkarte Instances (Instances).
3. Wählen Sie Remove from Load Balancer (Vom Load Balancer entfernen) in der Spalte Actions (Aktionen) der Instance aus.
4. Wenn Sie fertig sind, wählen Sie Edit Instances (Instances bearbeiten) aus und schicken die Instance an den Load Balancer zurück.

⚠ Important

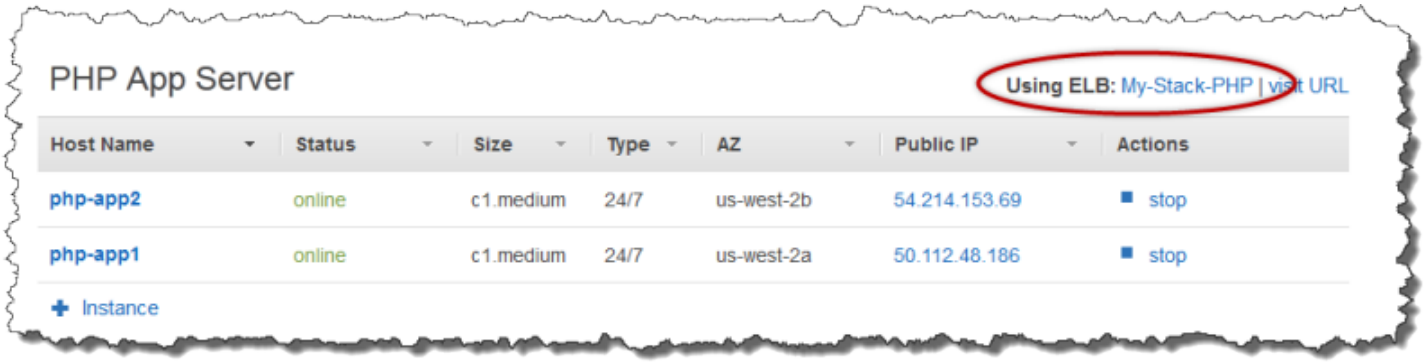
Wenn Sie die Elastic Load Balancing-Konsole oder API verwenden, um eine Instance aus einem Load Balancer zu entfernen, müssen Sie sie auch mit Elastic Load Balancing wiederherstellen. AWS OpsWorks Stacks ist sich der Operationen, die Sie mit anderen Servicekonsolen oder APIs ausführen, nicht bewusst und gibt die Instance nicht für Sie an den Load Balancer zurück. Manchmal kann AWS OpsWorks Stacks die Instanz wieder zum ELB hinzufügen, aber dieses Verhalten ist nicht garantiert und tritt nicht in allen Fällen auf.

Sie können mehrere Load Balancer an eine bestimmte Gruppe von Instances anfügen:

Anfügen mehrerer Load Balancer

1. Verwenden Sie die Elastic Load Balancing Balancing-Konsole, API oder CLI, um eine Reihe von Load Balancern zu erstellen.
2. Erstellen Sie einen benutzerspezifischen Layer für jeden Load Balancer und fügen Sie einen der Load Balancer an. Sie müssen keine benutzerspezifischen Rezepte für diese Layer implementieren, ein Standard-Layer ist ausreichend.
3. Fügen Sie die Gruppe der Instances jedem benutzerspezifischen Layer hinzu.

Sie können die Eigenschaften eines Load Balancers überprüfen, indem Sie die Seite "Instances" aufrufen und auf den Namen des entsprechenden Load Balancers klicken.



Die Seite ELB (ELB) zeigt die grundlegenden Eigenschaften des Load Balancers an, einschließlich seines DNS-Namens und des Zustandsprüfungsstatus der dazugehörigen Instances. Wenn der Stack in einem VPC ausgeführt wird, zeigt die Seite Subnetze statt der Availability Zones an. Ein grünes Häkchen verweist auf eine funktionierende Instance. Klicken Sie auf den Namen, um über den Load Balancer eine Verbindung mit einem Server herzustellen.

Amazon RDS-Serviceschicht

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Eine Amazon RDS-Serviceschicht stellt eine Amazon RDS-Instance dar. Die Ebene kann nur bestehende Amazon RDS-Instances darstellen, die Sie separat mithilfe der Amazon RDS-Konsole oder API erstellen müssen.

Das grundlegende Verfahren für die Integration einer Amazon RDS-Serviceschicht in Ihren Stack lautet wie folgt:

1. Verwenden Sie die Amazon RDS-Konsole, API oder CLI, um eine Instance zu erstellen.

   Stellen Sie sicher, dass Sie die Instance-ID, den Master-Benutzernamen, das Master-Passwort und den Datenbanknamen erfassen.

2. Um Ihrem Stack eine Amazon RDS-Ebene hinzuzufügen, registrieren Sie die Amazon RDS-Instance beim Stack.
3. Hängen Sie den Layer an eine App an, wodurch die Verbindungsinformationen der Amazon RDS-Instance zu den `deploy`Attributen der App hinzugefügt werden.
4. Verwenden Sie die sprachspezifischen Dateien oder die Informationen in den `deploy` Attributen, um die Anwendung mit der Amazon RDS-Instance zu verbinden.

   Weitere Informationen zum Verbinden einer Anwendung mit einem Datenbankserver finden Sie unter the section called “Verbinden mit einer Datenbank”

> **Warning**
>
> Stellen Sie sicher, dass die Zeichen, aus denen das Master-Passwort und der -Benutzername Ihrer Instance bestehen, mit Ihrem Anwendungsserver kompatibel sind. Wenn beispielsweise die Java App Server-Ebene & in einer der beiden Zeichenketten enthalten ist, wird ein XML-Analysefehler verursacht, der den Start des Tomcat-Servers verhindert.

Themen

Angeben von Sicherheitsgruppen

Um eine Amazon RDS-Instance mit AWS OpsWorks Stacks zu verwenden, müssen die Datenbank- oder VPC-Sicherheitsgruppen den Zugriff von den entsprechenden IP-Adressen aus zulassen. Für den Produktivbetrieb beschränkt die Sicherheitsgruppe in der Regel den Zugriff auf die IP-Adressen, die auf die Datenbank zugreifen müssen. Dazu gehören in der Regel die Adressen der Systeme, die Sie zur Verwaltung der Datenbank verwenden, und der AWS OpsWorks Stacks-Instances, die auf die Datenbank zugreifen müssen. AWS OpsWorks Stacks erstellt automatisch eine Amazon EC2-Sicherheitsgruppe für jeden Layer-Typ, wenn Sie Ihren ersten Stack in einer Region erstellen. Eine einfache Möglichkeit, Zugriff für AWS OpsWorks Stacks-Instances zu gewähren, besteht darin, der

Amazon RDS-Instance oder VPC die entsprechenden AWS OpsWorks Stacks-Sicherheitsgruppen zuzuweisen.

So geben Sie Sicherheitsgruppen für eine bestehende Amazon RDS-Instance an

1. Öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Klicken Sie im Navigationsbereich auf Instances und wählen Sie die entsprechende Amazon RDS-Instance aus. Klicken Sie auf Instance Actions (Instance-Aktionen), Modify (Ändern).
3. Wählen Sie die folgenden Sicherheitsgruppen aus der Liste Security Group (Sicherheitsgruppe) aus und klicken Sie dann auf Continue (Weiter) und Modify DB Instance (DB-Instance ändern), um die Instance zu aktualisieren.

   - *Die **Sicherheitsgruppe OpsWorks AWS-DB-Master-Server (security_group_id).***
   - Die Sicherheitsgruppe für den Anwendungsserver-Layer, dessen Instances eine Verbindung zur Datenbank aufnehmen. Der Gruppenname enthält den Layer-Namen. Um beispielsweise Datenbankzugriff auf PHP App Server-Instances bereitzustellen, geben Sie die Gruppe AWS-OpsWorks -PHP-App-Server an.

Wenn Sie eine neue Amazon RDS-Instance erstellen, können Sie die entsprechenden AWS OpsWorks Stacks-Sicherheitsgruppen auf der Seite „Erweiterte Einstellungen konfigurieren“ des Assistenten zum Starten einer DB-Instance angeben. Eine Beschreibung zur Verwendung dieses Assistenten finden Sie unter Erstellen einer MySQL DB-Instance und Verbinden mit einer Datenbank auf einer MySQL-DB-Instance.

Weitere Informationen zum Festlegen von VPC-Sicherheitsgruppen finden Sie unter Sicherheitsgruppen für Ihre VPC.

Registrierung einer Amazon RDS-Instance mit einem Stack

Um einen Amazon RDS-Service Layer zu einem Stack hinzuzufügen, müssen Sie eine Instance beim Stack registrieren.

Um eine Amazon RDS-Instance bei einem Stack zu registrieren

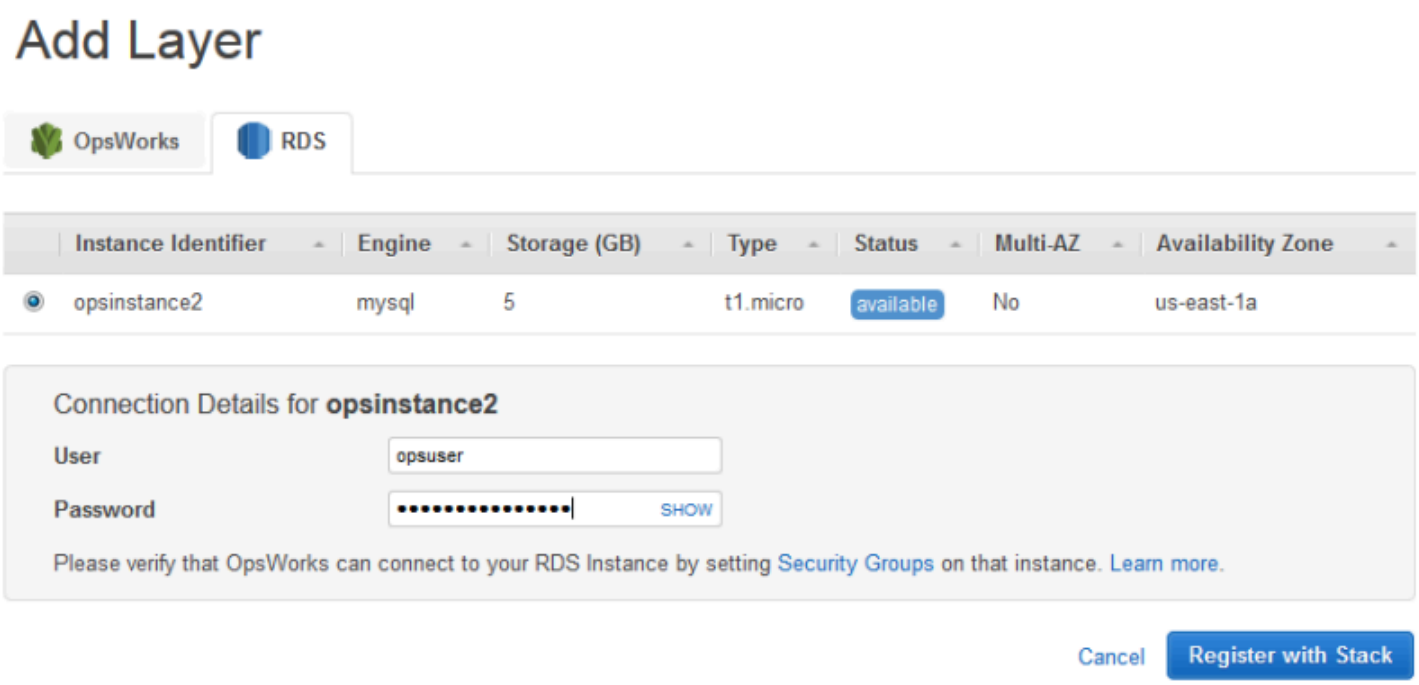
1. Klicken Sie in der AWS OpsWorks Stacks-Konsole im Navigationsbereich auf Layer, klicken Sie auf + Layer oder Layer hinzufügen, um die Seite „Layer hinzufügen“ zu öffnen, und klicken Sie dann auf die Registerkarte RDS.

2. Falls erforderlich, aktualisieren Sie die Stack-Servicerolle wie in Aktualisieren der Stack-Servicerolle beschrieben.
3. Klicken Sie auf die Registerkarte RDS, um die verfügbaren Amazon RDS-Instances aufzulisten.

> **Note**
>
> Wenn Ihr Konto keine Amazon RDS-Instances hat, können Sie eine erstellen, indem Sie auf der Registerkarte RDS auf RDS-Instance hinzufügen klicken. Dadurch gelangen Sie zur Amazon RDS-Konsole und starten den Assistenten zum Starten einer DB-Instance. Sie können auch direkt zur Amazon RDS-Konsole gehen und auf Eine DB-Instance starten klicken oder die Amazon RDS-API oder CLI verwenden. Weitere Informationen zum Erstellen einer Amazon RDS-Instance finden Sie unter Erste Schritte mit Amazon RDS.

4. Wählen Sie die geeignete Instance aus, tragen Sie bei User (Benutzer) und Password (Passwort) den richtigen Benutzer und das richtige Passwort ein und klicken Sie dann auf Register to Stack (Beim Stack registrieren).

> **Important**
>
> Sie müssen sicherstellen, dass der Benutzer und das Passwort, die Sie zur Registrierung der Amazon RDS-Instance verwenden, einem gültigen Benutzer und Passwort entsprechen. Ist dies nicht der Fall, können die Anwendungen keine Verbindung zur Instance herstellen. Sie können jedoch den Layer bearbeiten, um einen gültigen Benutzer und ein gültiges Passwort zu erstellen und dann die Anwendung erneut bereitstellen.



Wenn Sie einen Amazon RDS-Service Layer zu einem Stack hinzufügen, weist AWS OpsWorks Stacks ihm eine ID zu und fügt die zugehörige Amazon RDS-Konfiguration dem Attribut der Stack-Konfiguration und dem Attribut des Deployment-Attributs `[:opsworks][:stack]`hinzu.

> **Note**
>
> Wenn Sie das Passwort einer registrierten Amazon RDS-Instance ändern, müssen Sie das Passwort in AWS OpsWorks Stacks manuell aktualisieren und dann Ihre Apps erneut bereitstellen, um die Stack-Konfiguration und die Bereitstellungsattribute auf den Stack-Instances zu aktualisieren.

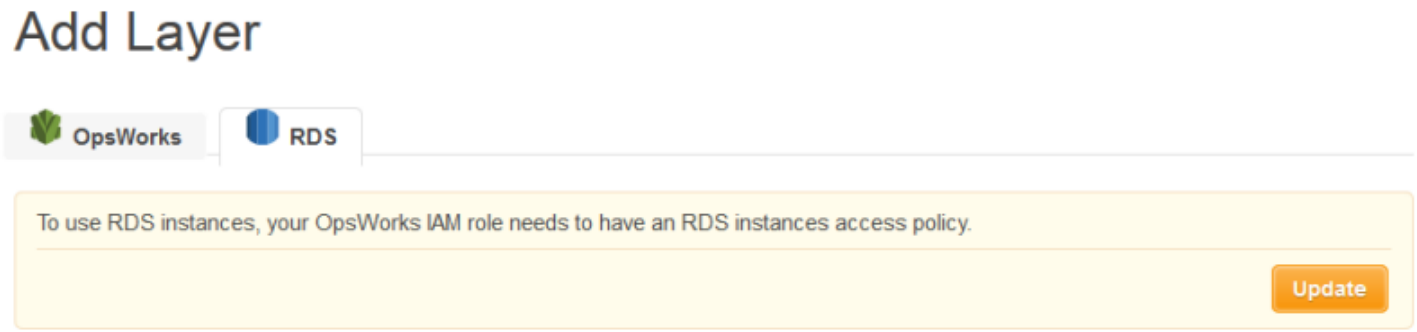
Themen

- Aktualisieren der Stack-Servicerolle

Aktualisieren der Stack-Servicerolle

Jeder Stack hat eine IAM-Servicerolle, die angibt, welche Aktionen AWS OpsWorks Stacks in Ihrem Namen mit anderen AWS-Services ausführen kann. Um eine Amazon RDS-Instance bei einem Stack zu registrieren, muss ihre Service-Rolle AWS OpsWorks Stacks Berechtigungen für den Zugriff auf Amazon RDS gewähren.

Wenn Sie zum ersten Mal einen Amazon RDS-Service-Layer zu einem Ihrer Stacks hinzufügen, fehlen der Service-Rolle möglicherweise die erforderlichen Berechtigungen. Wenn dies der Fall ist, sehen Sie Folgendes, wenn Sie auf die Registerkarte RDS auf der Seite Add Layer (layer hinzufügen) klicken.



Klicken Sie auf Aktualisieren, damit AWS OpsWorks Stacks die Richtlinie der Servicerolle wie folgt aktualisiert.

```
{"Statement": [{"Action": ["ec2:*", "iam:PassRole",
                      "cloudwatch:GetMetricStatistics",
                      "elasticloadbalancing:*",
                      "rds:*"],
            "Effect": "Allow",
            "Resource": ["*"] }]
}
```

> **Note**
>
> Sie müssen das Update nur einmal ausführen. Die aktualisierte Rolle wird dann automatisch von allen Ihren Stacks verwendet.

Amazon RDS Service Layers mit Apps verknüpfen

Nachdem Sie einen Amazon RDS-Service-Layer hinzugefügt haben, können Sie ihn mit einer App verknüpfen.

- Sie können einer App einen Amazon RDS-Layer zuordnen, wenn Sie die App erstellen, oder später, indem Sie die Konfiguration der App bearbeiten.

- Um eine Amazon RDS-Ebene von einer App zu trennen, bearbeiten Sie die Konfiguration der App, um einen anderen Datenbankserver oder keinen Server anzugeben.

  Die Amazon RDS-Ebene bleibt Teil des Stacks und kann mit einer anderen App verknüpft werden.

Nachdem Sie eine Amazon RDS-Instance mit einer App verknüpft haben, platziert AWS OpsWorks Stacks die Datenbankverbindungsinformationen auf den Servern der App. Diese Informationen können dann von der Anwendung auf jeder Server-Instance verwendet werden, um eine Verbindung mit der Datenbank herzustellen. Weitere Informationen zum Herstellen einer Verbindung mit einer Amazon RDS-Instance finden Sie unterthe section called “Verbinden mit einer Datenbank”.

Entfernen eines Amazon RDS-Service Layers aus einem Stack

Um einen Amazon RDS-Service Layer aus einem Stack zu entfernen, müssen Sie ihn deregistrieren.

So melden Sie einen Amazon RDS-Service Layer ab

1. Klicken Sie im Navigationsbereich auf Layers und dann auf den Namen des Amazon RDS-Service-Layers.
2. Klicken Sie auf Deregister (Abmelden) und bestätigen Sie, dass Sie den Layer abmelden möchten.

Dieses Verfahren entfernt die Ebene aus dem Stack, löscht jedoch nicht die zugrunde liegende Amazon RDS-Instance. Die Instance und alle Datenbanken verbleiben in Ihrem Konto und können mit anderen Stacks registriert werden. Sie müssen die Amazon RDS-Konsole, API oder CLI verwenden, um die Instance zu löschen. Weitere Informationen finden Sie unter Löschen einer DB-Instance.

ECS-Cluster-Ebenen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Der Amazon Elastic Container Service (Amazon ECS) verwaltet Docker-Container auf einem Cluster von Amazon Elastic Compute Cloud (Amazon EC2) -Instances, den sogenannten Container-Instances. Eine ECS-Cluster-Schicht stellt einen Amazon ECS-Cluster dar und vereinfacht die Clusterverwaltung, indem sie Funktionen wie die folgenden bietet:

- Optimierte Bereitstellung und Verwaltung der Container-Instance
- Betriebssystem und Paketaktualisierungen der Container-Instance
- Verwaltung von Benutzerberechtigungen
- Leistungsüberwachung der Container-Instance
- Volumenverwaltung für Amazon Elastic Block Store (Amazon EBS)
- Verwaltung von öffentlichen und Elastic IP-Adressen
- Verwaltung der Sicherheitsgruppe

Für die ECS-Cluster-Schicht gelten die folgenden Einschränkungen und Anforderungen:

- Der Layer ist nur für Chef 11.10- oder Chef 12-Linux-Stacks verfügbar, die in einer VPC ausgeführt werden, einschließlich einer Standard-VPC.
- Auf den Instances des Layers muss eines der folgenden Betriebssysteme ausgeführt werden.
  - 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
  - Benutzerdefiniert
- Die AWS OpsWorks Stacks-Agent-Version auf den Layer-Instances muss `3425-20150727112318` oder später sein.

Themen

Hinzufügen einer ECS-Clusterschicht zu einem Stack

AWS OpsWorks Stacks vereinfacht das Starten und Verwalten von Container-Instances für bestehende Amazon ECS-Cluster. Verwenden Sie die Amazon ECS-Konsole, die Befehlszeilenschnittstelle (CLI) oder die API, um andere Amazon ECS-Entitäten wie Cluster und Aufgaben zu erstellen oder zu starten. (Weitere Informationen finden Sie im Amazon Elastic Container Service Developer Guide.) Anschließend können Sie einem Stack einen Cluster zuordnen, indem Sie eine ECS-Cluster-Ebene erstellen, mit der Sie den Cluster in AWS OpsWorks Stacks verwalten können.

Sie können Cluster wie folgt mit Stacks verknüpfen:

- Jeder Stack kann eine ECS-Cluster-Schicht haben, die einen einzelnen Cluster darstellt.
- Ein Cluster kann mit nur einem Stack verknüpft werden.

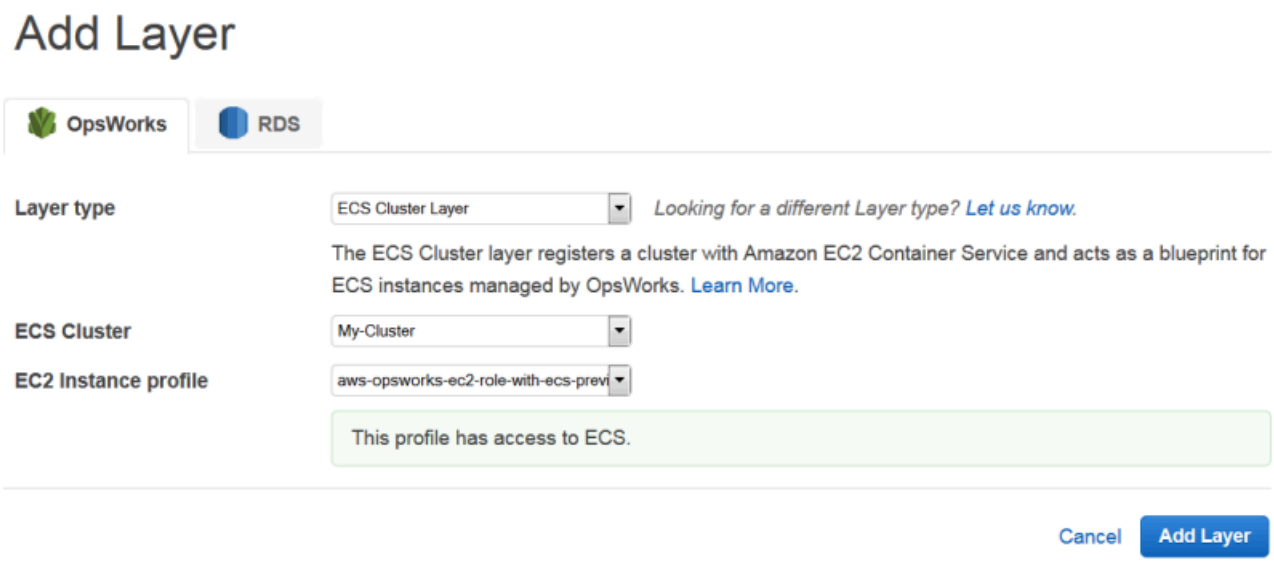
Bevor Sie ECS-Cluster-Layer zu Ihren Stacks hinzufügen können, müssen Sie die Servicerolle AWS OpsWorks Stacks AWS Identity and Access Management (IAM) aktualisieren, die normalerweise benannt ist`aws-opsworks-service-role`, damit AWS OpsWorks Stacks in Ihrem Namen mit Amazon ECS interagieren kann. Weitere Informationen zur Servicerolle finden Sie unter AWS OpsWorks Stacks erlauben, in Ihrem Namen zu handeln.

Wenn Sie zum ersten Mal eine ECS-Cluster-Ebene erstellen, bietet die Konsole eine Schaltfläche „Aktualisieren“, mit der Sie AWS OpsWorks Stacks anweisen können, die Rolle für Sie zu aktualisieren. AWS OpsWorks Stacks zeigt dann die Seite „Ebene hinzufügen“ an, auf der Sie die Ebene zum Stack hinzufügen können. Sie müssen die Servicerolle nur einmal aktualisieren. Anschließend können Sie die aktualisierte Rolle verwenden, um einem beliebigen Stack eine ECS-Cluster-Ebene hinzuzufügen.

> **Note**
>
> Wenn Sie möchten, können Sie die Richtlinie der Servicerolle auch manuell aktualisieren, indem Sie die Berechtigung `ecs:*` wie folgt zur vorhandenen Richtlinie hinzufügen:

```
{
  "Statement": [
    {
      "Action": [
        "ec2:*",
        "iam:PassRole",
        "cloudwatch:GetMetricStatistics",
        "elasticloadbalancing:*",
        "rds:*",
        "ecs:*"
      ],
      "Effect": "Allow",
      "Resource": ["*"]
    }
  ]
}
```

Für die Verknüpfung eines Clusters mit einem Stack sind zwei Vorgänge erforderlich: Registrieren des Clusters beim Stack und Erstellen des zugehörigen Layers. Die AWS OpsWorks Stacks-Konsole kombiniert diese Schritte. Bei der Layererstellung wird der angegebene Cluster automatisch registriert. Wenn Sie die AWS OpsWorks Stacks-API, CLI oder das SDK verwenden, müssen Sie separate Operationen verwenden, um den Cluster zu registrieren und die zugehörige Ebene zu erstellen. Um die Konsole zu verwenden, um Ihrem Stack eine ECS-Cluster-Ebene hinzuzufügen, wählen Sie Layers, wählen Sie +Layer oder Add a Layer und wählen Sie dann den Layer-Typ ECS-Cluster aus.



Auf der Seite Add Layer (Layer hinzufügen) stehen folgende Konfigurationsoptionen zur Verfügung:

ECS-Cluster

Der Amazon ECS-Cluster, den Sie beim Stack registrieren möchten.

EC2-Instance-Profil

Das Amazon Elastic Compute Cloud (Amazon EC2) Instance-Profil des Clusters. Dieses Profil gewährt Anwendungen, die auf den Container-Instances des Clusters ausgeführt werden, die Erlaubnis, auf andere AWS-Services, einschließlich Amazon ECS, zuzugreifen. Wenn Sie Ihre erste ECS-Cluster-Ebene erstellen, wählen Sie Neues Profil mit ECS-Zugriff, um AWS OpsWorks Stacks anzuweisen, das erforderliche Profil mit dem Namen `aws-opsworks-ec2-role-with-ecs` zu erstellen. Sie können dieses Profil dann für alle nachfolgenden ECS-Cluster-Ebenen verwenden. Weitere Informationen zum Instance-Profil finden Sie unter Festlegen von Berechtigungen für Apps auf EC2-Instances.

Sie können andere Einstellungen durch Bearbeiten der Layer-Konfiguration festlegen, darunter:

- Einen Elastic Load Balancing Load Balancer an die Ebene anhängen.

  Dieser Ansatz mag für einige Anwendungsfälle geeignet sein, Amazon ECS bietet jedoch anspruchsvollere Optionen. Weitere Informationen finden Sie unter Service Load Balancing.
- Festlegen, ob den Container-Instances automatisch öffentliche IP-Adressen oder Elastic IP-Adressen zugewiesen werden sollen.

Wenn Sie die automatische Zuweisung für beide Adresstypen deaktivieren, geht die Instance nicht online, es sei denn, das Subnetz besitzt einen ordnungsgemäß konfigurierten NAT. Weitere Informationen finden Sie unter Ausführen eines Stacks in einer VPC.

Verwalten des ECS-Clusters

Nachdem Sie eine ECS-Cluster-Ebene erstellt haben, können Sie AWS OpsWorks Stacks verwenden, um den Cluster wie folgt zu verwalten:

Bereitstellen und Verwalten von Container-Instances

Anfänglich umfasst eine ECS-Cluster-Ebene keine Container-Instances, selbst wenn dies im ursprünglichen Cluster der Fall war. Eine Möglichkeit besteht darin, die Instances des Layers durch eine geeignete Kombination der folgenden Verfahren zu verwalten:

- Fügen Sie 24/7-Instances manuell zum Layer hinzu und löschen Sie sie, wenn sie nicht mehr benötigt werden.
- Fügen Sie Instances zu einem Zeitplan hinzu bzw. löschen Sie sie, indem Sie zeitbasierte Instances zum Layer hinzufügen.
- Fügen Sie Instanzen auf der Grundlage von AWS OpsWorks Stacks-Host-Metriken oder CloudWatch -Alarmen hinzu oder löschen Sie sie, indem Sie der Ebene lastbasierte Instances hinzufügen.

> **Note**
>
> Wenn Amazon ECS für das Standardbetriebssystem des Stacks nicht unterstützt wird, müssen Sie explizit ein unterstütztes Betriebssystem angeben — 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 oder Benutzerdefiniert — wenn Sie erstellen Sie die Container-Instances. Verwenden Sie das ECS-optimierte AMI nicht, um Instances in einer ECS-Schicht zu erstellen, da dieses AMI den ECS-Agenten bereits enthält. AWS OpsWorks Stacks versucht außerdem, den ECS-Agenten während der Einrichtung der Instanz zu installieren, und der Konflikt kann dazu führen, dass die Installation fehlschlägt.

Weitere Informationen finden Sie unterOptimieren der Serveranzahl. AWS OpsWorks Stacks weist jeder Instance die OpsWorksAWS-ECS-Cluster-Sicherheitsgruppe zu. Nachdem jede neue Instance fertig gebootet wurde, konvertiert AWS OpsWorks Stacks sie in eine Container-Instance, indem Docker und der Amazon ECS-Agent installiert und die Instance anschließend im Cluster registriert werden.

Wenn Sie lieber vorhandene Container-Instances verwenden möchten, können Sie sie beim Stack registrieren und sie der ECS-Cluster-Ebene zuweisen. Beachten Sie, dass auf den Instances ein unterstütztes Betriebssystem ausgeführt werden muss: Amazon Linux 2015.03 oder höher oder Ubuntu 14.04 LTS oder höher.

> **Note**
>
> Eine Container-Instance kann nicht sowohl zu einer ECS-Cluster-Ebene als auch zu einer anderen integrierten Schicht gehören. Eine Container-Instance kann jedoch zu einer ECS-Cluster-Ebene und einer oder mehreren benutzerdefinierten Ebenen gehören.

Ausführen von Betriebssystem- und Paketaktualisierungen

Nachdem das Booten einer neuen Instance abgeschlossen ist, installiert AWS OpsWorks Stacks die neuesten Updates. Anschließend können Sie AWS OpsWorks Stacks verwenden, um die Container-Instances auf dem neuesten Stand zu halten. Weitere Informationen finden Sie unter Verwalten von Sicherheitsupdates.

Benutzerberechtigungen verwalten

AWS OpsWorks Stacks bietet eine einfache Möglichkeit, Berechtigungen für die Container-Instances zu verwalten, einschließlich der Verwaltung der SSH-Schlüssel der Benutzer. Weitere Informationen finden Sie unter Verwalten von Benutzerberechtigungen und Verwalten des SSH-Zugriffs.

Überwachen der Leistungskennzahlen

AWS OpsWorks Stacks bietet eine Vielzahl von Möglichkeiten, Leistungskennzahlen für den Stack, die Ebene oder einzelne Instances zu überwachen. Weitere Informationen finden Sie unter Überwachen.

Andere Verwaltungsaufgaben, wie das Erstellen von Aufgaben oder Services, erledigen Sie über Amazon ECS. Weitere Informationen finden Sie im Amazon Elastic Container Service-Entwicklerhandbuch.

> **Note**
>
> Um direkt zur Seite des Clusters in der Amazon ECS-Konsole zu gelangen, wählen Sie Instances und dann ECS-Cluster aus, was sich in der oberen rechten Ecke des Abschnitts der ECS-Cluster-Ebene befindet.

Löschen einer ECS-Cluster-Ebene aus einem Stack

Wenn Sie den Cluster nicht mehr benötigen, löschen Sie die ECS-Cluster-Schicht und melden Sie den zugehörigen Cluster ab. Für das Entfernen eines Clusters aus einem Stack sind zwei Vorgänge erforderlich: Abmelden des Clusters und Löschen des zugehörigen Layers. Die AWS OpsWorks Stacks-Konsole kombiniert diese Schritte. Durch das Löschen der Ebene wird der angegebene Cluster automatisch deregistriert. Wenn Sie die AWS OpsWorks Stacks-API, CLI oder das SDK verwenden, müssen Sie separate Operationen verwenden, um den Cluster abzumelden und die zugehörige Ebene zu löschen.

Um die Konsole zum Löschen einer ECS-Cluster-Ebene zu verwenden

1. Wenn Sie kontrollieren möchten, wie Aufgaben heruntergefahren werden, verwenden Sie die Amazon ECS-Konsole, API oder CLI, um die Dienste des Clusters herunterzuskalieren und zu löschen. Weitere Informationen finden Sie unter Aufräumen Ihrer Amazon ECS-Ressourcen.
2. Stoppen Sie die Instances des Layers und löschen Sie sie dann. Wenn Sie eine Container-Instance beenden, stoppt AWS OpsWorks Stacks automatisch alle laufenden Aufgaben, meldet die Instance vom Cluster ab und beendet die Instance.

   > **Note**
   >
   > Wenn Sie vorhandene Container-Instances beim Stack registriert haben, können Sie die Zuweisung der Instances vom Stack aufheben und sie dann abmelden, wodurch die Instances wieder durch ECS gesteuert werden.

3. Löschen Sie die Ebene. AWS OpsWorks Stacks hebt die Registrierung des zugehörigen Clusters auf, löscht ihn jedoch nicht. Der Cluster verbleibt in Amazon ECS.

Benutzerdefinierte AWS OpsWorks Stapel (Ebenen)

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Ein benutzerspezifischer Layer verfügt nur über eine minimale Anzahl an Rezepten. Anschließend fügen Sie dem Layer entsprechende Funktionen hinzu, indem Sie benutzerspezifische Rezepte implementieren und sie den Lebenszyklusereignissen des Layers zuordnen.

Der benutzerspezifische Layer hat die folgenden Konfigurationseinstellungen.

> **Note**
>
> AWS OpsWorks Stacks installiert Ruby automatisch auf den Instanzen der Ebene. Wenn Sie Ruby-Code auf der Instance ausführen möchten, aber nicht die Standard-Ruby-Version verwenden möchten, können Sie benutzerspezifisches JSON-Format oder benutzerspezifische Attributdateien verwenden, um die gewünschte Version anzugeben. Weitere Informationen finden Sie unter Ruby-Versionen.

Die grundlegenden Schritte zum Erstellen eines benutzerspezifischen Layers sind wie folgt:

1. Implementieren Sie ein Rezeptbuch, das die Rezepte und die zugeordneten Dateien enthält, die bei der Installation und Konfiguration von Paketen, bei der Bearbeitung von Konfigurationsänderungen, der Bereitstellung von Anwendungen etc. erforderlich sind.

   Je nach Ihren Anforderungen benötigen Sie unter Umständen auch Rezepte, die Bereitstellungen aufheben und Shutdowns ausführen. Weitere Informationen finden Sie unter Cookbooks und Rezepte.
2. Erstellen eines benutzerspezifischen Layers.
3. Weisen Sie Ihre Rezepte den entsprechenden Lebenszyklusereignissen zu.

Anschließend fügen Sie dem Layer die Instances hinzu, starten diese und stellen Anwendungen für diese Instances bereit.

> **Important**
>
> Um Anwendungen für Instances eines benutzerdefinierten Layers bereitzustellen, müssen Sie Rezepte implementieren, die Bereitstellungsvorgänge verarbeiten und sie dem Bereitstellungsereignis des Layers zuordnen.

Paketinstallationen für Ihr Betriebssystem pro Layer

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Ab Chef 12 müssen Sie benutzerdefinierte Rezepte verwenden, um Pakete auf Layer zu installieren, die auf verschiedenen Betriebssystemen ausgeführt werden. Auf diese Weise erhalten Sie maximale Flexibilität und Kontrolle über Paketinstallationen.

Nehmen wir beispielsweise an, Sie möchten Apache auf Layern installieren RedHat, auf denen Ubuntu- und Amazon-Versionen des Linux-Betriebssystems ausgeführt werden. Das Apache-Paket für RedHat und Amazon Linux heißt`httpd`, aber auf Ubuntu heißt es`apache2`.

Für unterschiedliche Paketbezeichnungen können Sie die Syntax wie im folgenden Beispielrezept verwenden. Mit dem Rezept wird das geeignete Apache-Paket für jedes Betriebssystem installiert. Dieses Beispiel basiert auf der Chef-Dokumentation.

```
package "Install Apache" do
   case node[:platform]
      when "redhat", "amazon"
         package_name "httpd"
      when "ubuntu"
         package_name "apache2"
```

```
    end
end
```

Detaillierte Informationen zur Verwendung der `package`-Ressource zum Verwalten von Paketen finden Sie in der Chef-Dokumentation auf der Seite Package.

Alternativ können Sie die Hilfsmethode `value_for_platform` von der Chef-Rezept-DSL (domänenspezifische Sprache) verwenden, mit der Sie schneller zum gleichen Ergebnis gelangen:

```
package "Install Apache" do
   package_name value_for_platform(
      ["redhat", "amazon"] => { "default" => "httpd" },
      ["ubuntu"] => { "default" => "apache2" }
   )
end
```

Weitere Informationen zur Verwendung der Hilfsmethode `value_for_platform` finden Sie unter About the Recipe DSL.

Instances

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Eine Instance stellt eine Rechenressource dar, z. B. eine Amazon EC2 EC2-Instance, die die Bereitstellung von Anwendungen, den Ausgleich des Datenverkehrs usw. übernimmt. Das Betriebssystem einer Instance kann eine Linux-Distribution oder Windows Server 2012 R2 sein.

Sie haben folgende Möglichkeiten, um einem Stack Instances hinzuzufügen:

- Verwenden Sie AWS OpsWorks Stacks, um Instances zu einem Stack hinzuzufügen. Die Instances, die Sie hinzufügen, stellen Amazon EC2 EC2-Instances dar.

- Für Linux-basierte Stacks können Sie Instances registrieren, die an anderer Stelle erstellt wurden — einschließlich Instances, die Sie mit Amazon EC2 erstellt haben, und On-Premises-Instances, die auf Ihrer eigenen Hardware ausgeführt werden.

  Anschließend können Sie AWS OpsWorks Stacks verwenden, um diese Instances auf die gleiche Weise zu verwalten wie mit Stacks erstellte Instances AWS OpsWorks

In diesem Abschnitt wird beschrieben, wie Sie AWS OpsWorks Stacks verwenden, um Instanzen zu erstellen und zu verwalten.

Themen

AWS OpsWorks Stacks-Instances verwenden

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Du kannst AWS OpsWorks Stacks verwenden, um Instanzen zu erstellen und sie dem Stack hinzuzufügen.

Themen

- Verwenden von benutzerdefinierten AMIs
- Manuelles Starten, Beenden und Neustarten von 24/7-Instances
- Verwaltung der Last mit zeit- und lastbasierten Instanzen

AWS OpsWorks Stacks-Betriebssysteme

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks unterstützt die 64-Bit-Versionen mehrerer integrierter Betriebssysteme, darunter Amazon- und Ubuntu-Linux-Distributionen sowie Microsoft Windows Server. Einige allgemeine Hinweise:

- Die Instances eines Stacks können mit Linux oder Windows ausgeführt werden.

  Ein Stack kann unterschiedliche Linux-Versionen oder -Distributionen auf verschiedenen Instances haben, es ist jedoch nicht möglich, Linux- und Windows-Instances zu kombinieren.
- Sie können benutzerdefinierte AMIs (Amazon Machine Images) verwenden, diese müssen jedoch auf einem der von AWS OpsWorks Stacks unterstützten AMIs basieren, die in den Themen dieses Abschnitts beschrieben werden. Es ist zwar möglich, Instances mit anderen Betriebssystemen (z. B. CentOS 6.x) zu erstellen oder zu registrieren, die mithilfe von benutzerdefinierten oder in der Community generierten AMIs erstellt wurden, offiziell werden diese Betriebssysteme jedoch nicht unterstützt.
  - Linux-Betriebssysteme
  - Microsoft Windows Server
- Sie können Instances manuell starten und stoppen oder die Anzahl der Instances durch AWS OpsWorks Stacks automatisch skalieren lassen.

  Sie können die zeitbasierte automatische Skalierung für jeden Stack verwenden, für Linux-Stacks kann auch die lastbasierte Skalierung eingesetzt werden.

- Neben der Verwendung von AWS OpsWorks Stacks zur Erstellung von Amazon EC2 EC2-Instances können Sie auch Instances mit einem Linux-Stack registrieren, die außerhalb von AWS OpsWorks Stacks erstellt wurden.

  Dazu gehören Amazon EC2 EC2-Instances und Instances, die auf Ihrer eigenen Hardware ausgeführt werden. Sie müssen jedoch eine der unterstützten Linux-Distributionen ausführen. Sie können keine Amazon EC2- oder lokalen Windows-Instances registrieren.

Sie können die AWS OpsWorks `DescribeOperatingSystems`Stacks-API ausführen, um eine Liste der unterstützten Betriebssysteme und ihrer unterstützten Versionen von Chef zurückzugeben. Im Folgenden finden Sie einen Beispielbefehl über die AWS CLI.

```
aws opsworks describe-operating-systems
```

Nachfolgend finden Sie eine Beispielantwort.

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

Themen

- Linux-Betriebssysteme
- Microsoft Windows Server

Linux-Betriebssysteme

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks unterstützt die 64-Bit-Versionen der folgenden Linux-Betriebssysteme.

- Amazon Linux und Amazon Linux 2 (die aktuell unterstützten Versionen finden Sie in der AWS OpsWorks Stacks-Konsole)
- Ubuntu 20.04 LTS
- CentOS 7
- Red Hat Enterprise Linux 7

Sie können auch benutzerdefinierte AMIs basierend auf diesen Betriebssystemen verwenden.

Einige allgemeine Hinweise zu Linux-Instances:

Unterstützte Paketversionen

Die unterstützten Versionen und Patch-Ebenen für Pakete wie Ruby hängen von dem Betriebssystem und der Version ab. Einzelheiten dazu finden Sie in den folgenden Abschnitten.

Aktualisierungen

Standardmäßig stellt AWS OpsWorks Stacks sicher, dass Linux-Instances über die neuesten Sicherheitspatches verfügen, indem es automatisch `yum update` oder `apt-get update` nach dem Start einer Instanz aufruft. Um automatische Updates zu deaktivieren, verwenden Sie die UpdateLayerAktionen CreateInstanceUpdateInstance, CreateLayer, oder — oder die entsprechenden AWS-SDK-Methoden oder AWS-CLI-Befehle —, um den `InstallUpdatesOnBoot` Parameter auf zu setzen. `false`

Um Serviceunterbrechungen zu vermeiden, installiert AWS OpsWorks Stacks Updates nicht automatisch, nachdem eine Instance online ist. Sie können das Betriebssystem einer Online-Instance jederzeit manuell aktualisieren, indem Sie den Stack-Befehl Upgrade Operating System ausführen. Weitere Informationen zum Verwalten von Sicherheitsaktualisierungen finden Sie unter Verwalten von Sicherheitsupdates.

Um mehr Kontrolle darüber zu haben, wie AWS OpsWorks Stacks Ihre Instances aktualisiert, erstellen Sie ein benutzerdefiniertes AMI, das auf einem der unterstützten Betriebssysteme basiert. Mit benutzerdefinierten AMIs können Sie beispielsweise angeben, welche Paketversionen

auf einer Instance installiert sind. Jede Linux-Distribution besitzt unterschiedliche Support-Zeitvorgaben und Richtlinien zum Zusammenführen von Paketen, daher sollten Sie sich überlegen, welche Methode am besten für Ihre Anforderungen geeignet ist. Weitere Informationen finden Sie unter Verwenden von benutzerdefinierten AMIs.

Hosts-Datei

Jede Online-Instanz hat eine `/etc/hosts` Datei, die IP-Adressen Hostnamen zuordnet. AWS OpsWorks Stacks enthält die öffentlichen und privaten Adressen für alle Online-Instanzen des Stacks in der `hosts` Datei jeder Instanz. Angenommen, Sie haben einen Stack mit zwei Node.js App Server-Instanzen, nodejs-app1 und nodejs-app2, und einer MySQL-Instanz, db-master1. Die `hosts`-Datei der Instance "nodejs-app1" sieht in etwa wie im folgenden Beispiel dargestellt aus. Die anderen Instances verfügen über ähnliche `hosts`-Dateien.

```
...
# 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 Unterstützung für Stacks Agent Proxy

Der AWS OpsWorks Stacks-Agent für Chef 11.10 und spätere Stacks bietet grundlegende Unterstützung für Proxyserver, die normalerweise mit isolierten VPCs verwendet werden. Um die Unterstützung für Proxy-Server aktivieren zu können, muss eine Instance eine Datei `/etc/environment` haben, die die entsprechenden Einstellungen für HTTP- und HTTPS-Datenverkehr enthält. Die Datei sollte wie im Folgenden dargestellt aussehen, wobei Sie den hervorgehobenen Text durch die URL und den Port Ihres Proxy-Servers ersetzen:

```
http_proxy="http://myproxy.example.com:8080/"
https_proxy="http://myproxy.example.com:8080/"
no_proxy="169.254.169.254"
```

Zum Aktivieren der Proxy-Unterstützung empfehlen wir, ein benutzerdefiniertes AMI zu erstellen, das eine entsprechende Datei `/etc/environment` enthält. Verwenden Sie dann dieses AMI zum Erstellen Ihrer Instances.

> **Note**
>
> Wir empfehlen nicht, ein benutzerdefiniertes Rezept zu verwenden, um eine `/etc/environment` Datei auf Ihren Instances zu erstellen. AWS OpsWorks Stacks benötigt die Proxy-Serverdaten zu einem frühen Zeitpunkt des Einrichtungsprozesses, bevor benutzerdefinierte Rezepte ausgeführt wurden.

Themen

- Amazon Linux
- Ubuntu LTS
- CentOS
- Red Hat Enterprise Linux

Amazon Linux

AWS OpsWorks Stacks unterstützt die 64-Bit-Versionen von Amazon Linux und Amazon Linux 2. Neben regelmäßigen Aktualisierungen und Patches erscheint etwa alle sechs Monate eine neue Amazon Linux-Version, die unter Umständen wesentliche Änderungen enthält. Bei der Erstellung eines Stacks oder einer neuen Instance müssen Sie angeben, welche Amazon Linux-Version verwendet werden soll. Wenn AWS eine neue Version veröffentlicht, führen Ihre Instances die angegebene Version weiterhin so lange aus, bis Sie sie explizit ändern. Nach der Veröffentlichung einer neuen Amazon Linux-Version gibt es einen 4-wöchigen Migrationszeitraum, in dem AWS weiterhin regelmäßige Aktualisierungen für die ältere Version bereitstellt. Nach diesem Migrationszeitraum können Ihre Instances weiterhin die ältere Version ausführen, jedoch stellt AWS keine weiteren Aktualisierungen bereit. Weitere Informationen finden Sie unter Amazon Linux AMI – Häufig gestellte Fragen.

Wenn eine neue Amazon Linux-Version veröffentlicht wird, empfehlen wir Ihnen, innerhalb des Migrationszeitraums auf die neue Version zu aktualisieren, damit Ihre Instances weiterhin Sicherheitsaktualisierungen erhalten. Vor der Aktualisierung der Instances Ihres Produktions-Stacks sollten Sie eine neue Instance starten und sich vergewissern, dass Ihre Anwendung einwandfrei

mit der neuen Version ausgeführt wird. Dann können Sie die Instances des Produktions-Stacks aktualisieren.

> **Note**
>
> Standardmäßig werden auf Amazon Linux basierte benutzerdefinierte AMIs automatisch auf die aktuelle Version aktualisiert, sobald diese veröffentlicht wurde. Es empfiehlt sich, Ihr benutzerdefiniertes AMI auf eine bestimmte Amazon Linux-Version zu beschränken. So können Sie Aktualisierungen zunächst testen, bevor Sie die neue Version verwenden. Weitere Informationen finden Sie unter Wie beschränke ich ein AMI auf eine bestimmte Version?.
>
> Wenn Sie eine AWS CloudFormation Vorlage verwenden, um Stacks mit Instances zu erstellen, auf denen Amazon Linux ausgeführt wird, sollten die Vorlagen explizit eine Amazon Linux-Version angeben. Das heißt, wenn in der Vorlage `Amazon Linux` angegeben ist, führen die Instances weiterhin Version 2016.09 aus. Weitere Informationen finden Sie unter AWS::OpsWorks::Stackund AWS::OpsWorks::Instance.

Zum Aktualisieren der Amazon Linux-Version einer Instance führen Sie einen der folgenden Schritte aus:

- Führen Sie bei Online-Instances den Stack-Befehl Upgrade Operating System aus.

  Wenn eine neue Amazon Linux-Version verfügbar ist, wird auf den Seiten Instances und Stack ein Hinweis mit einem Link angezeigt, über den Sie die Seite Run Command aufrufen. Sie können dann Upgrade Operating System ausführen, um ein Upgrade für Ihre Instance durchzuführen.
- Für Offline-Instances, die von Amazon Elastic Block Store (EBS-gestützt) unterstützt werden, starten Sie die Instances und führen Sie Upgrade Operating System aus, wie in der vorherigen Erklärung beschrieben.
- Für Instance-Speicher-gestützte Offline-Instances, einschließlich zeit- und lastbasierte Instances: Bearbeiten Sie die Einstellung Operating system der Instance, um die neue Version anzugeben.

  AWS OpsWorks Stacks aktualisiert die Instances automatisch auf die neue Version, wenn sie neu gestartet werden.

Amazon Linux: Unterstützte Node.js-Versionen

Amazon Linux-Version	Node.js-Versionen
2	(Not applicable to operating systems that are available for Chef 12 and higher stacks only)
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: Unterstützte Chef-Versionen

Chef-Version	Unterstützte Amazon Linux-Versionen
12	Amazon Linux 2 Amazon Linux 2018.03 Amazon Linux 2017.09 Amazon Linux 2017.03 Amazon Linux 2016.09

Chef-Version	Unterstützte Amazon Linux-Versionen
	Amazon Linux 2016.03
11.10	Amazon Linux 2018.03 Amazon Linux 2017.09 Amazon Linux 2017.03 Amazon Linux 2016.09 Amazon Linux 2016.03
11.4 (deprecated)	Amazon Linux 2016.09 Amazon Linux 2016.03

Important

Vergewissern Sie sich vor dem Aktualisieren von t1.micro-Instances, dass sie über die temporäre Auslagerungsdatei `/var/swapfile` verfügen. Die t1.micro-Instances auf Chef 0.9-Stacks haben keine Auslagerungsdatei. Auf Chef 11.4- und Chef 11.10-Stacks erstellen die aktuellen Versionen des Instance-Agenten automatisch eine Auslagerungsdatei für t1.micro-Instances. Diese Änderung wurde allerdings über einen Zeitraum von mehreren Wochen eingeführt, daher sollten Sie überprüfen, ob die Datei `/var/swapfile` auf Instances vorhanden ist, die vor dem 24. März 2014 erstellt wurden.
Für t1.micro-Instances, die keine Auslagerungsdatei haben, können Sie diese wie folgt erstellen:

- Für Chef 11.10-Stacks und höher: Erstellen Sie neue t1.micro-Instances, die automatisch eine Auslagerungsdatei aufweisen.
- Chef 0.9-Stacks: Führen Sie die folgenden Befehle auf jeder Instance als Root-Benutzer aus.

```
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
```

Sie können diese Befehle auch für Chef 11.10 und spätere Stacks verwenden, wenn Sie keine neuen Instances erstellen möchten.

Ubuntu LTS

Ubuntu veröffentlicht ungefähr alle zwei Jahre eine neue Ubuntu LTS-Version und bietet etwa fünf Jahre lang Unterstützung für die jeweilige Version. Ubuntu stellt Sicherheits-Patches und Sicherheitsaktualisierungen für die Dauer der Betriebssystemunterstützung bereit. Weitere Informationen finden Sie unter LTS – Ubuntu Wiki.

- Sie können eine vorhandene Ubuntu-Instance nicht auf eine neuere Version von Ubuntu aktualisieren.

  Sie müssen eine neue Ubuntu-Instanz erstellen und die ältere Instanz löschen.
- Ubuntu 20.04 LTS wird nur für Chef 12 und höhere Stacks unterstützt.

CentOS

AWS OpsWorks Stacks unterstützt die 64-Bit-Version von CentOS 7. Die ursprünglich unterstützte Version ist CentOS 7. CentOS veröffentlicht ungefähr alle zwei Jahre eine neue Version.

Wenn Sie eine neue Instanz in einem CentOS-Stack starten, installiert AWS OpsWorks Stacks automatisch die aktuellste CentOS-Version. Da AWS OpsWorks Stacks das Betriebssystem vorhandener Instanzen nicht automatisch aktualisiert, wenn eine neue CentOS-Nebenversion veröffentlicht wird, erhält eine neu erstellte Instanz möglicherweise eine neuere Version als die vorhandenen Instanzen des Stacks. Damit die Versionen auf Ihrem Stack einheitlich sind, können Sie Ihre vorhandenen Instances wie folgt auf die aktuelle CentOS-Version aktualisieren:

- Für Online-Instances: Um die Instances auf die aktuelle Version zu aktualisieren, führen Sie den Stack-Befehl Upgrade Operating System aus, mit dem `yum update` auf den angegebenen Instances ausgeführt wird.

  Wenn eine neue CentOS 7-Nebenversion verfügbar ist, wird auf den Seiten Instances und Stack ein Hinweis mit einem Link angezeigt, über den Sie die Seite Run Command aufrufen. Sie können dann Upgrade Operating System ausführen, um ein Upgrade für Ihre Instances durchzuführen.
- Für Offline-Instances, die von Amazon EBS unterstützt werden, starten Sie die Instances und führen Sie das Upgrade Operating System aus, wie im vorherigen Listenelement beschrieben.

- Bei Offline-Instances, die vom Store unterstützt werden, installiert AWS OpsWorks Stacks automatisch die neue Version, wenn die Instances neu gestartet werden.

CentOS: Unterstützte Chef-Versionen

Chef-Version	Unterstützte CentOS-Version
`12`	`CentOS 7`
`11.10`	`(None supported)`
`11.4 (deprecated)`	`(None supported)`

> **Note**
>
> AWS OpsWorks Stacks unterstützt Apache 2.4 für CentOS-Instanzen.

Red Hat Enterprise Linux

AWS OpsWorks Stacks unterstützt die 64-Bit-Version von Red Hat Enterprise Linux 7 (RHEL 7). Die ursprünglich unterstützte Version ist RHEL 7.1. Red Hat veröffentlicht ungefähr alle 9 Monate eine neue Nebenversion. Nebenversionen sollten mit RHEL 7.0 kompatibel sein. Weitere Informationen finden Sie unter Life Cycle and Update Policies.

Wenn Sie eine neue Instanz starten, installiert AWS OpsWorks Stacks automatisch die aktuelle RHEL 7-Version. Da AWS OpsWorks Stacks das Betriebssystem vorhandener Instanzen nicht automatisch aktualisiert, wenn eine neue RHEL 7-Nebenversion veröffentlicht wird, erhält eine neu erstellte Instanz möglicherweise eine neuere Version als die vorhandenen Instanzen des Stacks. Damit die Versionen auf Ihrem Stack einheitlich sind, können Sie Ihre vorhandenen Instances wie folgt auf die aktuelle RHEL 7-Version aktualisieren:

- Für Online-Instances: Um die Instances auf die aktuelle Version zu aktualisieren, führen Sie den Stack-Befehl Upgrade Operating System aus, mit dem `yum update` auf den angegebenen Instances ausgeführt wird.

Wenn eine neue RHEL 7-Version verfügbar ist, wird auf den Seiten Instances und Stack ein Hinweis mit einem Link angezeigt, über den Sie die Seite Run Command aufrufen. Sie können dann Upgrade Operating System ausführen, um ein Upgrade für Ihre Instances durchzuführen.

- Für Offline-Instances, die von Amazon EBS unterstützt werden, starten Sie die Instances und führen Sie das Upgrade Operating System aus, wie im vorherigen Listenelement beschrieben.
- Bei Offline-Instances, die vom Store unterstützt werden, installiert AWS OpsWorks Stacks automatisch die neue Version, wenn die Instances neu gestartet werden.

Red Hat Enterprise Linux: Unterstützte Node.js-Versionen

RHEL-Version	Node.js-Versionen
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: Unterstützte Chef-Versionen

Chef-Version	Unterstützte RHEL-Version
12	Red Hat Enterprise Linux 7
11.10	Red Hat Enterprise Linux 7

Chef-Version	Unterstützte RHEL-Version
`11.4 (deprecated)`	`(None supported)`

Alle Versionen von Node.js, die älter als 0.10.40 sind, sind veraltet. 0.12.7 und 0.12.9 sind ebenfalls veraltet.

> **Note**
>
> AWS OpsWorks Stacks unterstützt Apache 2.4 für RHEL 7-Instanzen.

Microsoft Windows Server

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In den folgenden Hinweisen wird die AWS OpsWorks Stacks-Unterstützung für Windows-Instanzen beschrieben. Windows-Instances sind nur für Chef 12.2-Stacks verfügbar. Die genaue Version von Chef in einem Windows-Stack ist 12.22.

Derzeit kann der AWS OpsWorks Stacks-Agent nicht auf Windows-basierten Instances installiert werden, die eine andere Sprache der Systembenutzeroberfläche als Englisch — USA (en-US) verwenden — und AWS OpsWorks Stacks kann diese auch nicht verwalten.

Versionen

AWS OpsWorks Stacks unterstützt die folgenden 64-Bit-Versionen von Windows:

- Microsoft Windows Server 2022-Basis
- Microsoft Windows Server 2022 mit SQL Server Express

- Microsoft Windows Server 2022 mit SQL Server-Standard
- Microsoft Windows Server 2022 mit SQL Server Web
- Microsoft Windows Server 2019-Basis
- Microsoft Windows Server 2019 mit SQL Server Express
- Microsoft Windows Server 2019 mit SQL Server Standard
- Microsoft Windows Server 2019 mit SQL Server Web

Erstellen von Instances

Sie erstellen Windows-Instanzen mit der AWS OpsWorks Stacks-Konsole, API oder CLI. Windows-Instances werden von Amazon EBS unterstützt, Sie können jedoch keine zusätzlichen Amazon EBS-Volumes mounten.

Windows-Stacks können 24/7-Instances verwenden, die Sie manuell starten und stoppen. Sie können auch das zeitbasierte Auto Scaling nutzen, mit dem die Instances basierend auf einem benutzerdefinierten Zeitplan automatisch gestartet und gestoppt werden. Windows-basierte Stacks können kein lastbasiertes Auto Scaling nutzen.

Sie können Windows-Instances, die außerhalb von AWS OpsWorks Stacks erstellt wurden, nicht mit einem Stack registrieren.

Aktualisierungen

AWS aktualisiert Windows-AMIs für den jeweiligen Satz von Patches, sodass die Instance bei der Erstellung stets über die neuesten Aktualisierungen verfügt. AWS OpsWorks Stacks bietet jedoch keine Möglichkeit, Updates auf Online-Windows-Instanzen anzuwenden. Die einfachste Methode, um sicherzustellen, dass Windows auf dem neuesten Stand ist, besteht darin, Ihre Instances regelmäßig zu ersetzen, damit diese immer mit dem neuesten AMI ausgeführt werden.

Ebenen

Zur Erledigung der Aufgaben wie Installieren und Konfigurieren von Software oder Bereitstellen von Anwendungen müssen Sie eine oder mehrere benutzerdefinierte Layer mit benutzerdefinierten Rezepten implementieren.

Chef

Windows-Instances verwenden Chef 12.22 und laufen im Chef-Client im lokalen Modus. Dieser startet einen lokalen In-Memory-Chef-Server mit dem Namen chef-zero. Dieser Server ermöglicht benutzerdefinierten Rezepten die Nutzung der Chef-Suchfunktion und Data Bags.

Remote-Anmeldung

AWS OpsWorks Stacks stellt autorisierten IAM-Benutzern ein Passwort zur Verfügung, mit dem sie sich bei Windows-Instanzen anmelden können. Dieses Passwort läuft nach einem bestimmten Zeitraum ab. Administratoren können mithilfe eines SSH-Schlüsselpaares das Administratorpasswort einer Instance abrufen, das uneingeschränkten RDP-Zugriff bietet. Weitere Informationen finden Sie unter Anmelden mit RDP.

AWS SDK

AWS OpsWorks Stacks installiert das automatisch AWS SDK for .NETauf jeder Instanz. Dieses Paket umfasst die AWS-.NET-Bibliotheken und AWS-Tools für Windows, einschließlich der AWS-Tools für PowerShell. Wenn Sie das Ruby-SDK verwenden möchten, können Sie das entsprechende Gem durch ein benutzerdefiniertes Rezept installieren lassen.

Überwachung und Metriken

Windows-Instances unterstützen die Amazon CloudWatch (CloudWatch) -Standardmetriken, die Sie in der CloudWatch Konsole anzeigen können.

Ruby

Der Chef 12.22-Client, den AWS OpsWorks Stacks auf Windows-Instanzen installiert, wird mit Ruby 2.3.6 geliefert. AWS OpsWorks Stacks fügt das Verzeichnis der ausführbaren Datei jedoch nicht zur Umgebungsvariablen PATH hinzu. Wenn Sie möchten, dass Ihre Anwendungen diese Ruby-Version verwenden, finden Sie sie typischerweise unter `C:\opscode\chef\embedded\bin\`.

AWS OpsWorks Stacks Agent CLI

Der AWS OpsWorks Stacks-Agent auf Windows-Instanzen stellt keine Befehlszeilenschnittstelle zur Verfügung.

Proxy-Unterstützung

Richten Sie die Proxy-Unterstützung für Windows-Instances wie folgt ein:

1. Ändern Sie die Änderung, `machine.config` um Folgendes hinzuzufügen, wodurch Proxyunterstützung für Windows- PowerShell (initialer Bootstrap) und .NET-Anwendungen (AWS OpsWorks Stacks-Agent) hinzugefügt wird:

```
<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. Führen Sie die folgenden Befehle aus, um die Umgebungsvariablen zu späteren Verwendung durch Chef und Git festzulegen:

```
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**
>
> Um mehr Kontrolle darüber zu haben, wie AWS OpsWorks Stacks Ihre Instances aktualisiert, erstellen Sie ein benutzerdefiniertes AMI auf Basis von Microsoft Windows Server 2022 Base. Mit benutzerdefinierten AMIs können Sie beispielsweise angeben, welche Software auf einer Instance installiert ist, z. B. als Web Server (IIS). Weitere Informationen finden Sie unter Verwenden von benutzerdefinierten AMIs.

Hinzufügen einer Instance zu einem Layer

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Nachdem Sie einen Layer erstellt haben, fügen Sie ihm in der Regel mindestens eine Instance hinzu. Falls die Auslastung steigt, können Sie jederzeit weitere Instances hinzufügen. Mithilfe von last- oder zeitbasierten Instances können Sie die Anzahl der Instances auch automatisch skalieren.

Sie können einem Layer sowohl neue als auch vorhandene Instances hinzufügen:

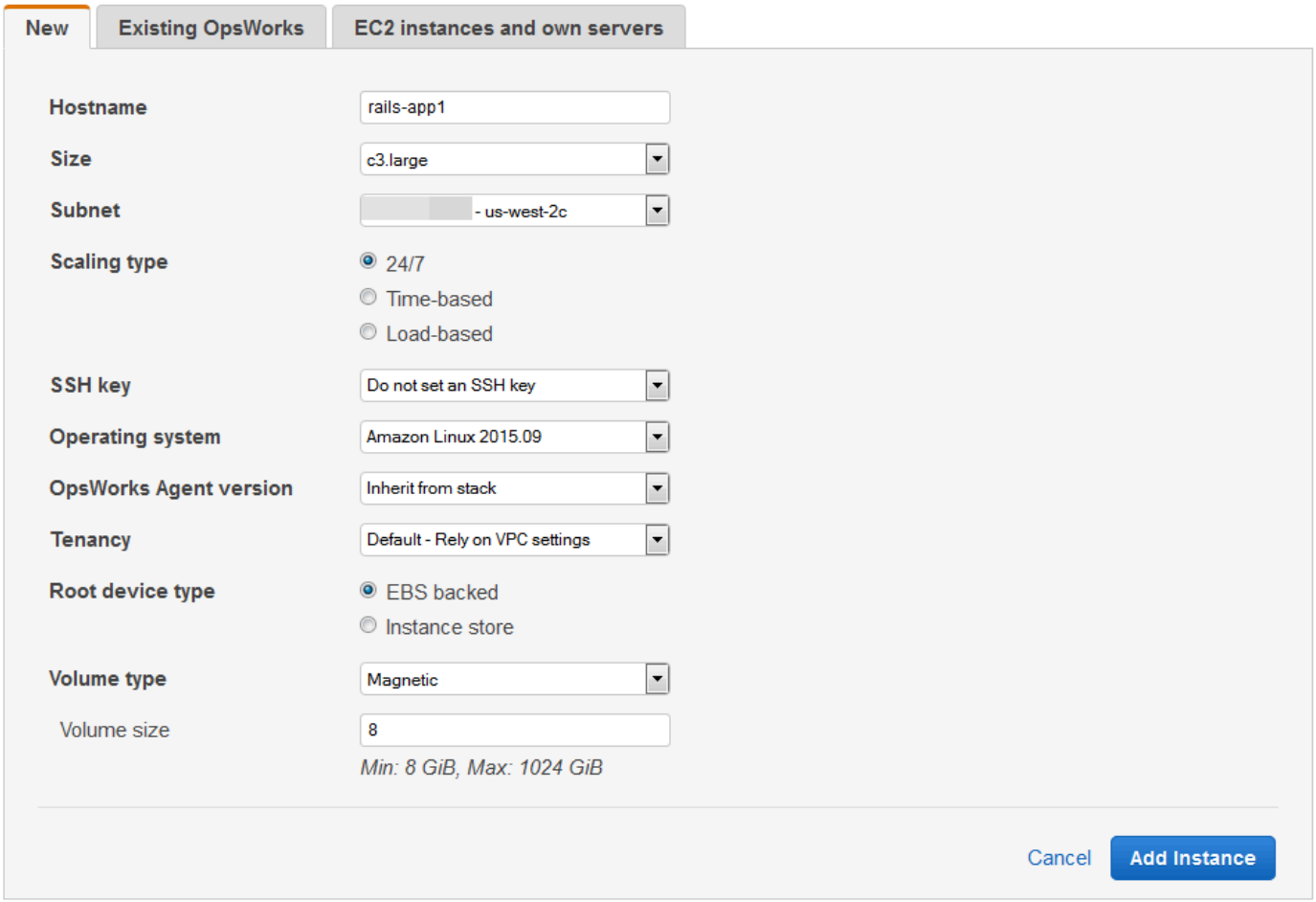
- Neu — OpsWorks erstellt eine neue Instanz, die nach Ihren Spezifikationen konfiguriert ist, und macht sie zu einem Mitglied des Layers.
- Existierend — Sie können eine vorhandene Instanz aus jedem kompatiblen Layer hinzufügen, sie muss sich jedoch im Offline-Status (gestoppt) befinden.

Wenn eine Instance mehreren Layers zugewiesen ist, führt AWS OpsWorks Stacks die Rezepte für jeden Layer der Instance aus, wenn ein Lebenszyklusereignis auftritt oder wenn Sie einen Stack-Befehl oder Bereitstellungs-Befehl ausführen.

Sie können eine Instance auch mehreren Layern zuweisen, indem Sie die Konfiguration der Instance bearbeiten. Weitere Informationen finden Sie unter Bearbeiten der Instance-Konfiguration.

So fügen Sie einem Layer eine neue Instance hinzu

1. Wählen Sie auf der Seite Instances die Option +Instance für den entsprechenden Layer und (gegebenenfalls) die Registerkarte New aus. Wenn Sie mehr als nur Host name, Size und Subnet oder Availability Zone konfigurieren möchten, wählen Sie Advanced >> aus, um weitere Optionen zu erhalten. Nachfolgend sehen Sie alle Optionen auf einen Blick:



2. Sie können die Standardkonfiguration, die Sie beim Erstellen des Stacks festgelegt haben, bei Bedarf überschreiben. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

   Hostname

   Identifiziert die Instance im Netzwerk. Standardmäßig generiert AWS OpsWorks Stacks den Hostnamen jeder Instanz mithilfe des Hostnamen-Designs, das Sie bei der Erstellung des Stacks angegeben haben. Sie können diesen Wert überschreiben und einen eigenen Hostnamen festlegen.

   Größe

   Ein Amazon EC2 EC2-Instance-Typ, der die Ressourcen der Instance spezifiziert, z. B. die Speichermenge oder die Anzahl der virtuellen Kerne. AWS OpsWorks Stacks gibt für jede Instance eine Standardgröße an, die Sie mit Ihrem bevorzugten Instance-Typ überschreiben können.

Die von AWS OpsWorks Stacks unterstützten Instance-Typen hängen davon ab, ob sich der Stack in einer VPC befindet oder nicht. Instance-Typen sind ebenfalls begrenzt, wenn Ihr Konto das kostenlosen Kontingent für AWS verwendet. Die Dropdown-Size-Liste zeigt die unterstützten Instance-Typen für die Chef-Version, die Ihr Stack unterstützt. Micro-Instances wie t1.micro verfügen möglicherweise nicht für alle Layer über ausreichende Ressourcen. Weitere Informationen finden Sie unter -Instance-Typen.

> **Note**
>
> Wenn Sie lastbasierte Instances verwenden, sollten Sie beachten, dass durch das Konfigurieren von Lebenszyklusereignissen möglicherweise die CPU-Auslastung eine Minute oder länger erheblich ansteigt. Bei kleinen Instances kann eine solche Auslastungsspitze bereits ausreichen, um eine Skalierung auszulösen, insbesondere bei großen, lastbasierten Stacks mit häufigen Konfigurationsereignissen. Nachfolgend werden einige Möglichkeiten beschrieben, mit denen Sie solche unnötigen Skalierungen vermeiden können.
>
> - Verwenden Sie größere Instances. So fällt die zusätzliche Auslastung durch Konfigurationsereignisse nicht so sehr ins Gewicht, dass eine Skalierung ausgelöst wird.
> - Verwenden Sie keine Instance-Typen wie T2, bei denen die CPU-Ressourcen geteilt werden.
>
>   So sind bei Konfigurationsereignissen die CPU-Ressourcen aller Instances sofort verfügbar.
> - Achten Sie darauf, dass die Zeitspanne für `exceeded threshold` deutlich über der Zeitspanne liegt, die für Konfigurationsereignisse nötig ist (z. B. 5 Minuten).
>
>   Weitere Informationen finden Sie unter Verwenden Sie die automatische lastbasierte Skalierung.

Availability Zone/Subnetz

Wenn der Stack nicht in einer VPC ausgeführt wird, heißt diese Einstellung Availability Zone und stellt die Zonen der Region dar. Mithilfe dieser Einstellung können Sie die Standard-Availability Zone, die Sie beim Erstellen des Stacks festgelegt haben, überschreiben.

Wenn der Stack in einer VPC ausgeführt wird, heißt diese Einstellung Subnet und listet die Subnetze der VPC auf. Mithilfe dieser Einstellung können Sie das Standard-Subnetz, das Sie beim Erstellen des Stacks festgelegt haben, überschreiben.

> **Note**
>
> Standardmäßig listet AWS OpsWorks Stacks die CIDR-Bereiche des Subnetzes auf. Um die Liste besser lesbar zu machen, verwenden Sie die VPC-Konsole oder API, um jedem Subnetz ein Tag hinzuzufügen, wobei Key auf **`Name`** und Value auf den Namen des Subnetzes gesetzt sind. AWS OpsWorks Stacks fügt diesen Namen an den CIDR-Bereich an. Im vorhergehenden Beispiel ist das Namens-Tag des Subnetzes **`Private`**.

Skalierungstyp

Legt fest, wie die Instance gestartet und angehalten wird.

- Der Standardwert ist eine 24/7-Instance, die Sie manuell starten und anhalten können.
- AWS OpsWorks Stacks startet und stoppt zeitbasierte Instances auf der Grundlage eines bestimmten Zeitplans.
- (Nur Linux) AWS OpsWorks Stacks startet und stoppt lastbasierte Instances auf der Grundlage bestimmter Lastmetriken.

> **Note**
>
> Last- bzw. zeitbasierte Instances werden nicht manuell gestartet und angehalten. Stattdessen konfigurieren Sie die Instances, und AWS OpsWorks Stacks startet und stoppt sie je nach Konfiguration. Weitere Informationen finden Sie unter Verwaltung der Last mit zeit- und lastbasierten Instanzen.

SSH-Schlüssel

Ein Amazon EC2 EC2-Schlüsselpaar. AWS OpsWorks Stacks installiert den öffentlichen Schlüssel auf der Instance.

- Für Linux-Instances können Sie den entsprechenden privaten Schlüssel mit einem SSH-Client verwenden, um sich bei der Instance anzumelden.
- Für Windows-Instances können Sie mithilfe des entsprechenden privaten Schlüssels das Administratorpasswort der Instance abrufen. Mit diesem Passwort wiederum können Sie sich über RDP als Administrator bei der Instance anmelden.

Zunächst ist diese Einstellung der Wert Default SSH key, den Sie beim Erstellen des Stacks festgelegt haben.

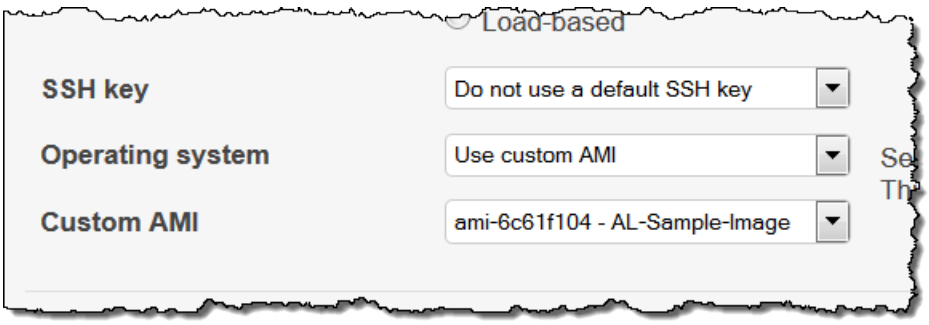
- Wenn der Standardwert auf Keinen Standard-SSH-Schlüssel verwenden gesetzt ist, können Sie einen der Amazon EC2 EC2-Schlüssel Ihres Kontos angeben.
- Wenn der Standardwert auf einen Amazon EC2 EC2-Schlüssel festgelegt ist, können Sie einen anderen Schlüssel oder keinen Schlüssel angeben.

Betriebssystem

Das Betriebssystem gibt an, auf welchem Betriebssystem die Instance ausgeführt wird. AWS OpsWorks Stacks unterstützt nur 64-Bit-Betriebssysteme.

Zunächst ist diese Einstellung der Wert Default operating system, den Sie beim Erstellen des Stacks festgelegt haben. Sie können den Standardwert überschreiben und ein anderes Linux-Betriebssystem oder ein benutzerdefiniertes Amazon Machine Image (AMI) festlegen. Es ist jedoch nicht möglich, zwischen Windows- und Linux-Betriebssystemen zu wechseln.

Wenn Sie Use custom AMI auswählen, zeigt die Seite eine Liste der benutzerdefinierten AMIs anstelle von Architecture und Root device type an.



Weitere Informationen finden Sie unter Verwenden von benutzerdefinierten AMIs.

OpsWorks Version des Agenten

OpsWorks Die Agentenversion gibt die Version des AWS OpsWorks Stacks-Agenten an, die Sie auf der Instanz ausführen möchten. Wenn AWS OpsWorks Stacks den

Agenten automatisch aktualisieren soll, wählen Sie Inherit Inherit from stack aus. Um eine bestimmte Version des Agenten zu installieren und den Agenten auf der Instance manuell zu aktualisieren, wählen Sie eine Version aus der Dropdown-Liste aus.

> **Note**
>
> Nicht alle Versionen des Agenten funktionieren mit allen Betriebssystemversionen. Wenn auf Ihrer Instance ein Agent ausgeführt wird — oder Sie einen Agenten auf einer Instance installieren —, der vom Instance-Betriebssystem nicht vollständig unterstützt wird, zeigt die AWS OpsWorks Stacks-Konsole Fehlermeldungen an, die Sie anweisen, einen kompatiblen Agenten zu installieren.

Tenancy

Wählen Sie die Mietoption für Ihre Instance aus. Sie können Ihre Instances wahlweise auf Servern ausführen, die ausschließlich für Sie reserviert sind.

- Default - Rely on VPC settings. Keine Miete, oder Mieteinstellungen werden von der VPC übernommen
- Dedicated - Run a dedicated instance. Stundenweise Abrechnung für Instances, die auf Einzelinstanzhardware ausgeführt werden. Weitere Informationen finden Sie unter Dedicated Instances im Amazon VPC-Benutzerhandbuch und unter Amazon EC2 Dedicated Instances.
- Dedicated host - Run this instance on a dedicated host. Sie zahlen für einen Host, der ausschließlich für Ihre Instances reserviert ist, und stellen eigene Softwarelizenzen pro Socket, Kern oder VM bereit, um Kosten zu sparen. Weitere Informationen finden Sie unter Übersicht über Dedicated Hosts in der Amazon EC2-Dokumentation und unter Amazon EC2 Dedicated Hosts.

Root-Gerätetyp

Legt den Root-Gerätespeicher der Instance fest.

- Linux-Instances können entweder Amazon EBS-gestützt oder Instance-Store-Backed sein.
- Windows-Instances müssen von Amazon EBS unterstützt werden.

Weitere Informationen hierzu finden Sie unter Speicher.

> **Note**
>
> Nach dem ersten Start booten Amazon EBS-gestützte Instances schneller als Instances Store-Backed Instances, da AWS OpsWorks Stacks die Software der Instance nicht von Grund auf neu installieren muss. Weitere Informationen finden Sie unter Root-Gerätespeicher.

Volume-Typ

Gibt den Typ des Root-Gerät-Datenträgers an: Magnetic, Provisioned IOPS (SSD) oder General Purpose (SSD). Weitere Informationen finden Sie unter Amazon EBS-Volume-Typen.

Volume-Größe

Legt die Root-Gerät-Volume-Größe für den festgelegten Volume-Typ fest. Weitere Informationen finden Sie unter Amazon EBS-Volume-Typen.

- Allzweck-SSD. Zulässige Mindestgröße: 8 GB, maximale Größe: 16 384 GB.
- Bereitgestellte IOPS (SSD) Zulässige Mindestgröße: 8 GB, maximale Größe: 16 384 GB. Für die Eingangs-/Ausgangsvorgänge pro Sekunde können Sie einen Wert zwischen 100 und 240 festlegen.
- Magnetic. Zulässige Mindestgröße: 8 GB, maximale Größe: 1 024 GB.

3. Wählen Sie Add Instance aus, um die neue Instance zu erstellen.

> **Note**
>
> Sie können die Standardversion des Stack-Agenten beim Erstellen einer Instance nicht überschreiben. Um eine benutzerdefinierte Agentenversion festzulegen, müssen Sie die Instance zunächst erstellen und dann die Konfiguration bearbeiten.

So fügen Sie einem Layer eine vorhandene Instance hinzu

1. Wählen Sie auf der Seite Instances die Option +Instance für den entsprechenden Layer aus und öffnen Sie dann die Registerkarte Existing.

> **Note**
>
> Wenn Sie doch lieber eine neue Instance erstellen möchten, wählen Sie New aus, um wie vorher beschrieben eine neue Instance zu erstellen.

2. Wählen Sie auf der Registerkarte Existing eine Instance aus der Liste aus.
3. Wählen Sie Add Instance aus, um die neue Instance zu erstellen.

Eine Instance stellt eine Amazon EC2 EC2-Instance dar, ist aber im Grunde nur eine AWS OpsWorks Stacks-Datenstruktur. Eine Instance muss gestartet werden, um eine laufende Amazon EC2 EC2-Instance zu erstellen, wie in den folgenden Abschnitten beschrieben.

> **Important**
>
> Wenn Sie eine Instance in einer Standard-VPC starten, müssen Sie beim Ändern der VPC-Konfiguration vorsichtig vorgehen. Die Instances müssen immer in der Lage sein, mit dem AWS OpsWorks Stacks-Service, Amazon S3 und Paket-Repositorys zu kommunizieren. Wenn Sie beispielsweise ein Standard-Gateway entfernen, verlieren die Instances ihre Verbindung zum AWS OpsWorks Stacks-Dienst, der die Instances dann als ausgefallen behandelt und sie auto repariert. AWS OpsWorks Stacks ist jedoch nicht in der Lage, den Instance-Agenten auf den reparierten Instances zu installieren. Ohne Agent können die Instances nicht mit dem Service kommunizieren und bleiben beim Hochfahren beim Status `booting` hängen. Weitere Informationen über die Standard-VPC finden Sie unter Unterstützte Plattformen.

Sie können auch Linux-Rechenressourcen in einen Stack integrieren, die außerhalb von AWS OpsWorks Stacks erstellt wurden:

- Amazon EC2 EC2-Instances, die Sie direkt mit der Amazon EC2 EC2-Konsole, CLI oder API erstellt haben.
- Lokale Instances, die auf Ihrer eigenen Hardware ausgeführt werden, einschließlich Instances auf virtuellen Maschinen.

Weitere Informationen finden Sie unter Verwenden von Computing-Ressourcen, die nicht mit AWS OpsWorks Stacks erstellt wurden.

Verwenden von benutzerdefinierten AMIs

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks unterstützt zwei Möglichkeiten zur Anpassung von Instances: benutzerdefinierte Amazon Machine Images (AMIs) und Chef-Rezepte. Mit beiden Ansätzen haben Sie die Kontrolle darüber, welche Pakete und Paketversionen installiert werden, wie diese konfiguriert werden usw. Jeder der beiden Ansätze bietet jedoch eigene Vorteile und die Wahl des richtigen Ansatzes hängt von Ihren Anforderungen ab.

Nachfolgend finden Sie die Hauptgründe für die Verwendung eines benutzerdefinierten AMIs:

- Sie möchten bestimmte Pakete vorab zusammenstellen, statt sie nach dem Hochfahren der Instance zu installieren.
- Sie möchten den Zeitpunkt von Paketaktualisierungen kontrollieren, um ein konsistentes Basisabbild für Ihren Layer bereitzustellen.
- Sie möchten Instances, insbesondere lastbasierte Instances so schnell wie möglich hochfahren.

Nachfolgend finden Sie die Hauptgründe für die Verwendung von Chef-Rezepten:

- Sie sind flexibler als benutzerdefinierte AMIs.
- Sie lassen sich einfacher aktualisieren.
- Sie können Online-Instances aktualisieren.

In der Praxis ist die optimale Lösung möglicherweise eine Kombination aus beiden Ansätzen. Weitere Informationen zu Rezepten finden Sie unter Cookbooks und Rezepte.

Themen

- So funktionieren benutzerdefinierte AMIs mit Stacks AWS OpsWorks

- Ein benutzerdefiniertes AMI für AWS OpsWorks Stacks erstellen

So funktionieren benutzerdefinierte AMIs mit Stacks AWS OpsWorks

Um ein benutzerdefiniertes AMI für Ihre Instances anzugeben, wählen Sie Benutzerdefiniertes AMI als Betriebssystem der Instance verwenden, wenn Sie eine neue Instance erstellen. AWS OpsWorks Stacks zeigt dann eine Liste der benutzerdefinierten AMIs in der Region des Stacks an, und Sie wählen die entsprechende AMIs aus der Liste aus. Weitere Informationen finden Sie unter Hinzufügen einer Instance zu einem Layer.

> **Note**
>
> Sie können kein bestimmtes benutzerdefiniertes AMI als Standardbetriebssystem eines Stacks festlegen. Sie können `Use custom AMI` als Standardbetriebssystem des Stacks festlegen, es ist jedoch nur beim Hinzufügen neuer Instances zu einem Layer möglich, ein bestimmtes AMI auszuwählen. Weitere Informationen finden Sie unter Hinzufügen einer Instance zu einem Layer und Erstellen eines neuen Stacks. Es ist zwar möglich, Instances mit anderen Betriebssystemen (z. B. CentOS 6.x) zu erstellen, die mithilfe von benutzerdefinierten oder in der Community generierten AMIs erstellt wurden, offiziell werden diese Betriebssysteme jedoch nicht unterstützt.

In diesem Thema werden einige allgemeine Probleme angesprochen, die Sie vor dem Erstellen oder Verwenden von benutzerdefinierten AMIs berücksichtigen sollten.

Themen

- Verhalten beim Hochfahren
- Auswählen eines Layers
- Umgang mit Anwendungen

Verhalten beim Hochfahren

Wenn Sie die Instance starten, verwendet AWS OpsWorks Stacks das angegebene benutzerdefinierte AMI, um eine neue Amazon EC2 EC2-Instance zu starten. AWS OpsWorks Stacks verwendet dann cloud-init, um den AWS OpsWorks Stacks-Agenten auf der Instance zu installieren, und der Agent führt die Setup-Rezepte der Instance aus, gefolgt von den Deploy-

Rezepten. Nachdem die Instance online ist, führt der Agent die Konfigurationsrezepte für jede Instance im Stack einschließlich der neu hinzugefügten Instance aus.

Auswählen eines Layers

Der AWS OpsWorks Stacks-Agent steht normalerweise nicht in Konflikt mit installierten Paketen. Die Instanz muss jedoch mindestens einer Ebene angehören. AWS OpsWorks Stacks führt immer die Rezepte dieser Ebene aus, was zu Problemen führen kann. Sie müssen genau verstehen, was die Rezepte eines Layers auf einer Instance tun, bevor Sie diesem Layer eine Instance mit einem benutzerdefinierten AMI hinzufügen.

Um zu prüfen, welche Rezepte ein bestimmter Layer-Typ auf Ihrer Instance ausführt, öffnen Sie einen Stack, der diesen Layer enthält. Klicken Sie dann im Navigationsbereich auf Layers und anschließend auf Recipes für den gewünschten Layer. Klicken Sie auf den Rezeptnamen, um den eigentlichen Code anzuzeigen.

> **Note**
>
> Bei Linux-AMIs besteht eine Möglichkeit, die Wahrscheinlichkeit von Konflikten zu verringern, darin, AWS OpsWorks Stacks zur Bereitstellung und Konfiguration der Instance zu verwenden, die die Grundlage für Ihr benutzerdefiniertes AMI bildet. Weitere Informationen finden Sie unter Erstellen Sie ein benutzerdefiniertes Linux-AMI aus einer AWS OpsWorks Stacks-Instance.

Umgang mit Anwendungen

Neben Paketen möchten Sie möglicherweise auch eine Anwendung in das AMI aufnehmen. Bei großen, komplexen Anwendungen kann sich die Zeit zum Hochfahren der Instance verkürzen, wenn Sie die Anwendung in das AMI aufnehmen. Sie können kleine Anwendungen in Ihr AMI aufnehmen, aber im Vergleich zur Bereitstellung der Anwendung durch AWS OpsWorks Stacks bietet das Bereitstellen der Anwendung in der Regel nur einen geringen oder gar keinen Zeitvorteil.

Eine Möglichkeit besteht darin, die Anwendung in Ihr AMI aufzunehmen und zusätzlich eine App zu erstellen, die die Anwendung über ein Repository auf den Instances bereitstellt. So lässt sich nicht nur die Startzeit verkürzen, es ist auch eine praktische Möglichkeit, die Anwendung nach dem Start der Instance zu aktualisieren. Beachten Sie, dass Chef-Rezepte idempotent sind. Bereitstellungsrezepte nehmen daher keine Änderungen an der Anwendung vor, solange die Version im Repository mit der auf der Instance übereinstimmt.

Ein benutzerdefiniertes AMI für AWS OpsWorks Stacks erstellen

Um ein benutzerdefiniertes AMI mit AWS OpsWorks Stacks zu verwenden, müssen Sie zunächst ein AMI aus einer benutzerdefinierten Instance erstellen. Sie können aus zwei Optionen wählen:

- Verwenden Sie die Amazon EC2 EC2-Konsole oder API, um eine Instance zu erstellen und anzupassen, die auf einer 64-Bit-Version eines der von AWS OpsWorks Stacks unterstützten AMIs basiert.
- Verwenden Sie für Linux-AMIs, OpsWorks um eine Amazon EC2 EC2-Instance auf der Grundlage der Konfiguration der zugehörigen Ebenen zu erstellen.

Bevor Sie ein benutzerdefiniertes Linux-AMI erstellen, deaktivieren Sie es `noexec` auf der `/tmp` Partition, damit AWS OpsWorks Stacks seinen Agenten auf benutzerdefinierten Linux-Instances installieren kann.

> **Note**
>
> Ein AMI funktioniert möglicherweise nicht auf allen Instance-Typen. Sie sollten daher sicherstellen, dass Ihr AMI mit den zu verwendenden Instance-Typen kompatibel ist. Insbesondere die R3-Instance-Typen benötigen ein AMI mit hardwaregestützter Virtualisierung.

Anschließend verwenden Sie die Amazon EC2 EC2-Konsole oder API, um aus der benutzerdefinierten Instance ein benutzerdefiniertes AMI zu erstellen. Sie können benutzerdefinierte AMIs in allen Stacks in derselben Region verwenden. Fügen Sie dazu einem Layer eine Instance hinzu und legen Sie Ihr benutzerdefiniertes AMI fest. Weitere Informationen dazu, wie Sie eine Instance erstellen, die ein benutzerdefiniertes AMI verwendet, finden Sie unter Hinzufügen einer Instance zu einem Layer.

> **Note**
>
> Standardmäßig installiert AWS OpsWorks Stacks alle Amazon Linux-Updates beim Booten, sodass Sie die neueste Version erhalten. Außerdem erscheint etwa alle sechs Monate eine neue Amazon Linux-Version, die unter Umständen wichtige Änderungen enthält. Standardmäßig werden auf Amazon Linux basierte benutzerdefinierte AMIs automatisch auf die aktuelle Version aktualisiert, sobald diese veröffentlicht wurde. Es empfiehlt sich, Ihr benutzerdefiniertes AMI auf eine bestimmte Amazon Linux-Version zu beschränken.

So können Sie Aktualisierungen zunächst testen, bevor Sie die neue Version verwenden. Weitere Informationen finden Sie unter Wie beschränke ich ein AMI auf eine bestimmte Version?.

Themen

- Erstellen Sie ein benutzerdefiniertes AMI mit Amazon EC2
- Erstellen Sie ein benutzerdefiniertes Linux-AMI aus einer AWS OpsWorks Stacks-Instance
- Erstellen eines benutzerdefinierten Windows-Amazon-Computerabbilds (AMI)

Erstellen Sie ein benutzerdefiniertes AMI mit Amazon EC2

Die einfachste Methode, ein benutzerdefiniertes AMI zu erstellen — und die einzige Option für Windows-AMIs — besteht darin, die gesamte Aufgabe mithilfe der Amazon EC2 EC2-Konsole oder API auszuführen. Weitere Informationen zu den folgenden Schritten finden Sie unter Erstellen eigener AMIs.

Um ein benutzerdefiniertes AMI mit der Amazon EC2 EC2-Konsole oder API zu erstellen

1. Erstellen Sie eine Instance auf Basis einer 64-Bit-Version eines von AWS OpsWorks Stacks unterstützten AMIs.
2. Passen Sie die Instance aus Schritt 1 an, indem Sie sie konfigurieren, Pakete installieren usw. Alles, was Sie installieren, wird auf sämtlichen auf diesem AMI basierenden Instances nachgebildet. Installieren Sie daher nichts, was nur auf dieser Instance laufen soll.
3. Halten Sie die Instance an und erstellen Sie ein benutzerdefiniertes AMI.

Erstellen Sie ein benutzerdefiniertes Linux-AMI aus einer AWS OpsWorks Stacks-Instance

Um eine benutzerdefinierte AWS OpsWorks Stacks Linux-Instance zur Erstellung eines AMI zu verwenden, beachten Sie, dass jede von erstellte Amazon EC2 EC2-Instance eine eindeutige Identität OpsWorks enthält. Wenn Sie aus einer solchen Instance ein benutzerdefiniertes AMI erstellen, enthält es diese Identität, und alle auf dem AMI basierenden Instances haben dieselbe Identität. Damit jede auf Ihrem benutzerdefinierten AMI basierende Instance eine eindeutige Identität hat, müssen Sie die Identität vor dem Erstellen des AMIs aus der angepassten Instance entfernen.

So erstellen Sie ein benutzerdefiniertes AMI aus einer AWS OpsWorks Stacks-Instance

1. Erstellen Sie einen Linux-Stack und fügen Sie mindestens einen Layer hinzu, um die Konfiguration der angepassten Instance festzulegen. Sie können sowohl integrierte Layers an Ihre Bedürfnisse anpassen als auch völlig eigene Layers verwenden. Weitere Informationen finden Sie unter Stacks anpassen AWS OpsWorks.
2. Bearbeiten Sie die Ebenen und deaktivieren Sie AutoHealing sie.
3. Fügen Sie den Layers eine Instance mit Ihrer bevorzugten Linux-Distribution hinzu und starten Sie sie. Wir empfehlen die Verwendung einer Amazon EBS-gestützten Instance. Öffnen Sie die Detailseite der Instance und notieren Sie sich ihre Amazon EC2 EC2-ID für später.
4. Nachdem die Instance online ist, melden Sie sich mit SSH an und führen abhängig vom Betriebssystem Ihrer Instance einen der nächsten vier Schritte aus.
5. Für eine Amazon Linux-Instance in einem Chef 11- oder Chef 12-Stack oder eine Red Hat Enterprise Linux 7-Instance in einem Chef 11-Stack gehen Sie wie folgt vor.

   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**
   >
   > Fügen Sie für Instances in einem Chef 12-Stack die folgenden beiden Verzeichnisse zu diesem Befehl hinzu:
   >
   > - `/var/chef`
   > - `/opt/chef`

   d. `sudo rpm -e opsworks-agent-ruby`

   e. `sudo rpm -e chef`

6. Bei einer Ubuntu 16.04 oder 18.04 LTS-Instance in einem Chef 12-Stack gehen Sie wie folgt vor.

   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. Für andere unterstützte Ubuntu-Versionen in einem Chef 12-Stack gehen Sie wie folgt vor.

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. Für eine Red Hat Enterprise Linux 7-Instance in einem Chef 12-Stack gehen Sie wie folgt vor.

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. Dieser Schritt ist abhängig vom Instance-Typ:

- Verwenden Sie für eine Amazon EBS-gestützte Instance die AWS OpsWorks Stacks-Konsole, um die Instance zu beenden und das AMI zu erstellen, wie unter Erstellen eines Amazon EBS-gestützten Linux-AMI beschrieben.

- Erstellen Sie für eine Instance Store-Backed Instance das AMI wie unter Erstellen eines Instance Store-Backed Linux AMI beschrieben und beenden Sie die Instance dann mit der AWS OpsWorks Stacks-Konsole.

  Fügen Sie beim Erstellen des AMIs unbedingt auch die Zertifikatdateien ein. Rufen Sie beispielsweise den Befehl `ec2-bundle-vol` mit dem Argument `-i` mit den Optionen `-i $(find /etc /usr /opt -name '*.pem' -o -name '*.crt' -o -name '*.gpg' | tr '\n' ',')` auf. Entfernen Sie beim Bündeln nicht die öffentlichen Schlüssel. Diese Aufgabe erledigen Sie mit dem Standardbefehl `ec2-bundle-vol`.

10. Bereinigen Sie Ihren Stack, indem Sie zur AWS OpsWorks Stacks-Konsole zurückkehren und die Instance aus dem Stack löschen.

Erstellen eines benutzerdefinierten Windows-Amazon-Computerabbilds (AMI)

Mit den folgenden Verfahren werden benutzerdefinierte AMIs für Windows Server 2022 Base erstellt. In der Amazon EC2-Managementkonsole können Sie andere Windows Server-Betriebssysteme auswählen.

> **Important**
>
> Derzeit kann der AWS OpsWorks Stacks-Agent nicht auf Windows-basierten Instances installiert werden, die eine andere Sprache der Systembenutzeroberfläche als Englisch — USA (en-US) verwenden — und AWS OpsWorks Stacks kann diese auch nicht verwalten.

Themen

- Erstellen eines benutzerdefinierten Windows-AMIs mit Sysprep
- Erstellen eines benutzerdefinierten Windows-AMIs ohne Sysprep
- Hinzufügen einer neuen Instance mithilfe eines benutzerdefinierten Windows-AMIs

Erstellen eines benutzerdefinierten Windows-AMIs mit **`Sysprep`**

Ein mit Sysprep erstelltes benutzerdefiniertes Windows-AMI führt zwar in der Regel zu einem langsameren Startprozess der Instance, ist allgemein jedoch sauberer. Der erstmalige Start einer Instanz, die aus einem mit erstellten Image erstellt wurde, `Sysprep` nimmt aufgrund von `Sysprep` Aktivitäten, Neustarts, der AWS OpsWorks Stacks-Bereitstellung und der ersten Ausführung von

Stacks, einschließlich Einrichtung und Konfiguration, mehr Zeit in Anspruch. AWS OpsWorks Führen Sie die Schritte zur Erstellung eines benutzerdefinierten Windows-AMI in der Amazon EC2 EC2-Konsole aus.

So erstellen Sie ein benutzerdefiniertes Windows-AMI mit Sysprep:

1. Wählen Sie in der Amazon-EC2-Konsole Instance starten aus.
2. Suchen Sie nach Microsoft Windows Server 2022 Base, und wählen Sie dann Auswählen aus.
3. Wählen Sie den gewünschten Instance-Typ aus und wählen Sie dann Next: Configure Instance Details aus. Passen Sie die Konfiguration des AMIs einschließlich Computername, Speicher- und Sicherheitsgruppeneinstellungen an. Wählen Sie Launch (Starten) aus.
4. Nachdem der Startprozess der Instance abgeschlossen ist, rufen Sie Ihr Passwort ab und melden Sie sich in einem Remote Desktop Connection-Fenster von Windows bei der Instance an.
5. Wählen Sie auf dem Windows-Startbildschirm Start und beginnen Sie dann mit der Eingabe, **`ec2configservice`** bis die Ergebnisse in der ConfigServiceSettingsEC2-Konsole angezeigt werden. Öffnen Sie die -Konsole.
6. Vergewissern Sie sich, dass auf der Registerkarte Allgemein das Kontrollkästchen UserData Ausführung aktivieren aktiviert ist (diese Option ist zwar nicht erforderlich für`Sysprep`, aber erforderlich, damit AWS OpsWorks Stacks seinen Agenten installiert). Deaktivieren Sie das Kontrollkästchen für die Option Set the computer name of the instance... (Computername der Instance einrichten), da diese Option zu einer Neustartschleife von AWS OpsWorks Stacks führen kann.
7. Legen Sie auf der Registerkarte Image das Administratorkennwort entweder auf Random fest, damit Amazon EC2 automatisch ein Passwort generieren kann, das Sie mit einem SSH-Schlüssel abrufen können, oder auf Specify, um Ihr eigenes Passwort anzugeben. `Sysprep`speichert diese Einstellung. Wenn Sie ein eigenes Passwort festlegen, speichern Sie sich dieses Passwort ab. Wir empfehlen Ihnen, Keep Existing nicht auszuwählen.
8. Wählen Sie Apply und anschließend Shutdown with Sysprep aus. Wenn Sie aufgefordert werden, Ihre Entscheidung zu bestätigen, wählen Sie Yes aus.
9. Nachdem die Instance gestoppt wurde, klicken Sie in der Amazon EC2 EC2-Konsole mit der rechten Maustaste auf die Instance in der Instance-Liste, wählen Sie Image und dann Create Image.
10. Geben Sie auf der Seite Create Image einen Namen und eine Beschreibung für das Abbild ein und legen Sie die Volume-Konfiguration fest. Wählen Sie Create Image aus, wenn Sie fertig sind.

11. Öffnen Sie die Seite Images und warten Sie, bis der Status des Abbilds von pending zu available wechselt. Das neue AMI ist nun einsatzbereit.

Erstellen eines benutzerdefinierten Windows-AMIs ohne **`Sysprep`**

Führen Sie die Schritte zur Erstellung eines benutzerdefinierten Windows-AMI in der Amazon EC2 EC2-Konsole aus.

So erstellen Sie ein benutzerdefiniertes Windows-AMI ohne Sysprep

1. Wählen Sie in der Amazon-EC2-Konsole Instance starten aus.
2. Suchen Sie nach Microsoft Windows Server 2022 Base, und wählen Sie dann Auswählen aus.
3. Wählen Sie den gewünschten Instance-Typ aus und wählen Sie dann Next: Configure Instance Details aus. Passen Sie die Konfiguration des AMIs einschließlich Computername, Speicher- und Sicherheitsgruppeneinstellungen an. Wählen Sie Launch (Starten) aus.
4. Nachdem der Startprozess der Instance abgeschlossen ist, rufen Sie Ihr Passwort ab und melden Sie sich in einem Remote Desktop Connection-Fenster von Windows bei der Instance an.
5. Öffnen Sie auf der Instance `C:\Program Files\Amazon\Ec2ConfigService\Settings\config.xml`, ändern Sie die folgenden beiden Einstellungen und speichern und schließen Sie dann die Datei:
   - `Ec2SetPassword` auf `Enabled`
   - `Ec2HandleUserData` auf `Enabled`
6. Trennen Sie die Verbindung zur Remote Desktop-Sitzung und kehren Sie zur Amazon EC2 EC2-Konsole zurück.
7. Halten Sie in der Liste Instances die Instance an.
8. Nachdem die Instance gestoppt wurde, klicken Sie in der Amazon EC2 EC2-Konsole mit der rechten Maustaste auf die Instance in der Instance-Liste, wählen Sie Image und dann Create Image.
9. Geben Sie auf der Seite Create Image einen Namen und eine Beschreibung für das Abbild ein und legen Sie die Volume-Konfiguration fest. Wählen Sie Create Image aus, wenn Sie fertig sind.
10. Öffnen Sie die Seite Images und warten Sie, bis der Status des Abbilds von pending zu available wechselt. Das neue AMI ist nun einsatzbereit.

Hinzufügen einer neuen Instance mithilfe eines benutzerdefinierten Windows-AMIs

Nachdem Ihr Abbild den Status available anzeigt, können Sie basierend auf Ihrem benutzerdefinierten Windows-AMI neue Instances erstellen. Wenn Sie Use custom Windows AMI aus der Liste Operating system auswählen, zeigt AWS OpsWorks Stacks eine Liste benutzerdefinierter AMIs an.

So fügen Sie eine neue Instance basierend auf einem benutzerdefinierten Windows-AMI hinzu

1. Wenn Ihr neues AMI verfügbar ist, gehen Sie zur AWS OpsWorks Stacks-Konsole, öffnen Sie die Instance-Seite für einen Windows-Stack und wählen Sie unten auf der Seite + Instance aus, um eine neue Instance hinzuzufügen.
2. Wählen Sie auf der Registerkarte New (Neu) die Option Advanced (Erweitert) aus.
3. Wählen Sie in der Dropdown-Liste Operating system die Option Use custom Windows AMI aus.
4. Wählen Sie in der Dropdown-Liste Custom AMI das erstellte AMI und anschließend Add Instance aus.

Sie können die Instance jetzt starten und ausführen.

Manuelles Starten, Beenden und Neustarten von 24/7-Instances

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Sie können 24/7-Instances mit Linux- und Windows-Stacks verwenden.

Nachdem Sie einer Ebene eine 24/7-Instance hinzugefügt haben, müssen Sie die Instance manuell starten, um die entsprechende Amazon Elastic Compute Cloud (Amazon EC2) -Instance zu starten, und sie manuell beenden, um die Amazon EC2 EC2-Instance zu beenden. Sie können Instances,

die nicht ordnungsgemäß funktionieren, auch manuell neu starten. AWS OpsWorks Stacks startet und stoppt automatisch zeit- und lastbasierte Instances. Weitere Informationen finden Sie unter Verwaltung der Last mit zeit- und lastbasierten Instanzen.

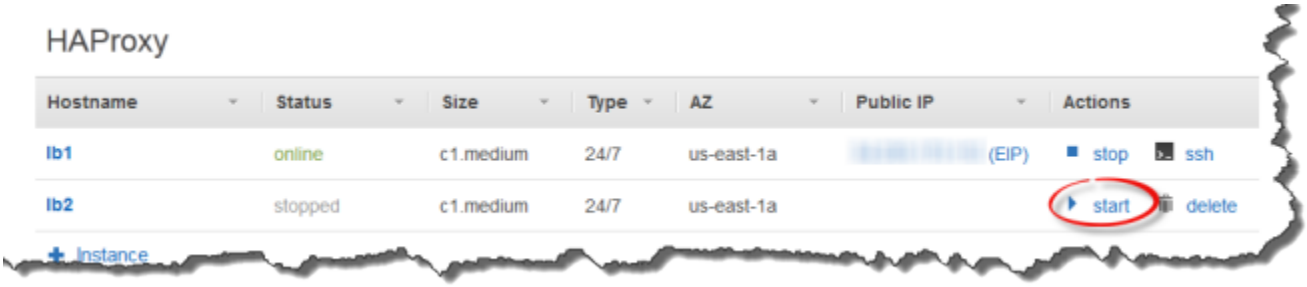
> **Important**
>
> AWS OpsWorks Stacks-Instanzen dürfen nur in der Konsole gestartet, gestoppt und neu gestartet werden. AWS OpsWorks AWS OpsWorks erkennt keine Start-, Stopp- oder Neustartvorgänge, die in der Amazon EC2 EC2-Konsole ausgeführt werden.

Themen

- Starten oder Neustarten einer Instance
- Anhalten einer Instance
- Neustarten einer Instance

Starten oder Neustarten einer Instance

Um eine neue Instance auf der Seite Instances zu starten, klicken Sie auf start in der Spalte Actions der Instance.



Sie können auch mehrere Instances erstellen und sie alle gleichzeitig starten, indem Sie auf Start all Instances klicken.

Nachdem Sie die Instance gestartet haben, startet AWS OpsWorks Stacks eine Amazon EC2 EC2-Instance und bootet das Betriebssystem. Der Startvorgang dauert in der Regel wenige Minuten und ist normalerweise für Windows-Instances etwas langsamer als für Linux-Instances. Während des Startprozesses zeigt das Feld Status der Instance folgende Werte an:

1. angefordert — AWS OpsWorks Stacks hat den Amazon EC2-Service aufgerufen, um die Amazon EC2 EC2-Instance zu erstellen.

2. ausstehend — AWS OpsWorks Stacks wartet auf den Start der Amazon EC2 EC2-Instance.
3. booten — Die Amazon EC2 EC2-Instance bootet.
4. running_setup — AWS OpsWorks Stacks hat das Setup-Ereignis ausgelöst und führt die Rezepte der Ebene aus, gefolgt von Setup ihren Rezepten. Deploy Weitere Informationen finden Sie unter Ausführen von Rezepten. Wenn Sie benutzerdefinierte Kochbücher zum Stapel hinzugefügt haben, installiert AWS OpsWorks Stacks die aktuelle Version aus Ihrem Repository, bevor die Rezepte und -Rezepte ausgeführt werden. Setup Deploy
5. online – Die Instance ist bereit zur Nutzung.

Wenn sich der Status in online ändert, ist die Instance vollständig betriebsbereit.

- Wenn der Layer über einen angehängten Load Balancer verfügt, fügt AWS OpsWorks Stacks die Instanz hinzu.
- AWS OpsWorks Stacks löst ein Configure Ereignis aus, das die Rezepte jeder Instanz ausführt. Configure

  Wie benötigt, aktualisieren diese Rezepte die Instance, um die neue Instance zu unterstützen.
- AWS OpsWorks Stacks ersetzt die Startaktion der Instanz durch die Stoppaktion, mit der Sie die Instanz beenden können.

Wenn die Instance nicht erfolgreich gestartet wurde oder die Einrichtungsrezepte fehlgeschlagen sind, wird der Status auf start_failed oder setup_failed gesetzt. Sie können die Protokolle überprüfen, um die Ursache zu ermitteln. Weitere Informationen finden Sie unter Handbuch zur Fehlersuche und Fehlerbehebung.

Eine gestoppte Instance bleibt Teil des Stacks und behält alle Ressourcen bei. Beispielsweise sind Amazon EBS-Volumes und Elastic IP-Adressen immer noch mit einer gestoppten Instance verknüpft. Sie können eine angehaltene Instance neu starten, indem Sie start in der Spalte Actions der Instance auswählen. Das Neustarten einer angehaltenen Instance bewirkt Folgendes:

- Instance Store-Backed Instances — AWS OpsWorks Stacks startet eine neue Amazon EC2 EC2-Instance mit derselben Konfiguration.
- Amazon EBS-gestützte Instances — AWS OpsWorks Stacks startet die Amazon EC2 EC2-Instance neu, wodurch das Root-Volume erneut angehängt wird.

Nachdem der Start der Instance abgeschlossen ist, installiert AWS OpsWorks Stacks Betriebssystem-Updates und führt die und -Rezepte aus, genau wie beim ersten Setup Start. Deploy AWS OpsWorks Stacks führt bei neu gestarteten Instanzen je nach Bedarf auch die folgenden Schritte aus.

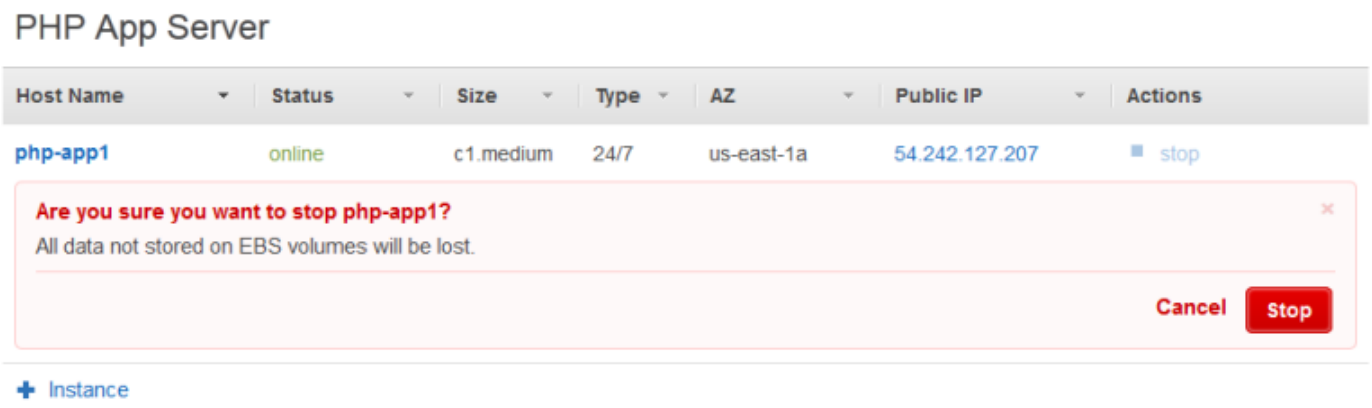
- Weist Elastic IP-Adressen neu zu.
- Hängt Amazon Elastic Block Store (Amazon EBS) -Volumes erneut an.
- Installiert die neuesten Rezeptbuch-Versionen für Instance-Speicher-gestützte Instances.

  Amazon EBS-gestützte Instances verwenden weiterhin die benutzerdefinierten Kochbücher, die auf dem Root-Volume gespeichert wurden. Wenn sich Ihre benutzerdefinierten Rezeptbücher verändert haben, seit Sie die Instance angehalten haben, müssen Sie sie manuell aktualisieren, nachdem die Instance online ist. Weitere Informationen finden Sie unter Aktualisieren von benutzerdefinierten Rezeptbüchern.

> **Note**
>
> Es kann einige Minuten dauern, bis eine Elastic IP-Adresse wieder einer neugestarteten Instance zugewiesen ist. Beachten Sie, dass die Elastic IP-Einstellung der Instance die Metadaten repräsentiert und einfach darauf hinweist, dass die Adresse mit der Instance verknüpft sein soll. Die Public IP-Einstellung spiegelt den Status der Instance wider und kann zunächst leer sein. Wenn die Elastic IP-Adresse der Instance zugewiesen ist, wird die Adresse der Public IP-Einstellung zugeordnet, gefolgt von (EIP).

Anhalten einer Instance

Klicken Sie auf der Seite Instances in der Spalte Aktionen der Instance auf Stopp. Dadurch wird AWS OpsWorks Stacks aufgefordert, die Shutdown-Rezepte auszuführen und die EC2-Instance zu beenden.



Sie können auch jede Instance in dem Stack abschalten, indem Sie auf Stop All Instances klicken.

Nachdem Sie die Instance gestoppt haben, führt AWS OpsWorks Stacks mehrere Aufgaben aus:

1. Wenn der Layer der Instance über einen Elastic Load Balancing Load Balancer verfügt, hebt AWS OpsWorks Stacks die Registrierung der Instance auf.

   Wenn die Ebene den Verbindungsausgleich des Load Balancers unterstützt, verzögert AWS OpsWorks Stacks das Auslösen des Shutdown-Ereignisses, bis der Verbindungsausgleich abgeschlossen ist. Weitere Informationen finden Sie unter Elastic Load Balancing Lastenausgleichsebene.
2. AWS OpsWorks Stacks löst ein Shutdown Ereignis aus, das die Rezepte der Instance ausführt. Shutdown
3. Nach dem Auslösen des Shutdown Ereignisses wartet AWS OpsWorks Stacks eine bestimmte Zeit ab, bis die Shutdown Rezepte fertig sind, und führt dann Folgendes aus:
   - Beendet Instance-Speicher-gestützte Instances, wodurch alle Daten gelöscht werden.
   - Stoppt Amazon EBS-gestützte Instances, wodurch die Daten auf dem Root-Volume erhalten bleiben.

   Weitere Informationen zum Instance-Speicher finden Sie unter Storage.

> **Note**
>
> Die Standardeinstellung für den Shutdown-Timeout ist 120 Sekunden. Wenn Ihre Shutdown-Rezepte mehr Zeit benötigen, können Sie die Layer-Konfiguration bearbeiten, um die Einstellung zu ändern.

Sie können den Shutdown-Prozess überwachen, indem Sie die Spalte Status der Instance beobachten. Während der Shutdown-Prozess verläuft, werden die folgenden Werte angezeigt:

1. terminierend — AWS OpsWorks Stacks beendet die Amazon EC2 EC2-Instance.
2. shutting_down — AWS OpsWorks Stacks führt die Rezepte der Ebene aus. Shutdown
3. beendet — Die Amazon EC2 EC2-Instance ist beendet.
4. stopped – Die Instance wurde angehalten.

Neustarten einer Instance

Klicken Sie auf der Seite Instances auf die nicht funktionierenden Instance-Namen, um die Detailseite anzuzeigen, und klicken Sie dann auf Reboot.

Dieser Befehl führt einen Soft-Neustart der zugehörigen Amazon EC2 EC2-Instance durch. Dadurch werden die Daten der Instance, selbst bei Instance-Speicher-gestützten Instances, nicht gelöscht und es werden keine Lebenszyklusereignisse ausgelöst.

> **Note**
>
> Damit AWS OpsWorks Stacks ausgefallene Instances automatisch ersetzen, aktivieren Sie Auto Healing. Weitere Informationen finden Sie unter Verwenden von Auto Healing.

Verwaltung der Last mit zeit- und lastbasierten Instanzen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Da Ihr eingehender Datenverkehr variiert, hat Ihr Stack entweder zu wenige Instances, um die Last problemlos zu verarbeiten, oder mehr Instances als nötig. Sie sparen Zeit und Geld, indem Sie zeit- oder lastbasierte Instances benutzen, um die Instances eines Layers automatisch zu erhöhen oder zu verringern, sodass Sie immer genug Instances haben, um eingehenden Datenverkehr ordnungsgemäß zu verarbeiten, ohne dass Kosten für nicht benötigte Kapazität entstehen. Es ist nicht notwendig, Serverlasten zu überwachen oder Instances manuell zu starten oder anzuhalten. Zeit- und lastbasierte Instances vertreiben, skalieren und stimmen Anwendungen zusätzlich über mehrere Availability Zones innerhalb einer Region automatisch ab. Dadurch bekommen Sie geografische Redundanz und Skalierbarkeit.

Die automatische Skalierung basiert auf zwei Instance-Typen, die die Online-Instances eines Layers, basierend auf verschiedenen Kriterien, anpassen:

- Time-based-Instances

  Sie erlauben einem Stack, Lasten zu verarbeiten, die einem vorhersehbaren Muster folgen, indem Sie Instances, die nur zu bestimmten Zeitpunkten oder an bestimmten Tagen ausgeführt werden, einschließen. Sie können beispielsweise einige Instances nach 18 Uhr starten, um nächtliche Sicherungsaufgaben auszuführen, oder einige Instances an Wochenenden anhalten, wenn weniger Datenverkehr stattfindet.
- Load-based-Instances

  Einem Stack wird erlaubt, variable Lasten zu verarbeiten, indem zusätzliche Instances bei hohem Datenaufkommen gestartet und Instances bei niedrigem Datenaufkommen angehalten werden. Dies basiert auf allen Auslastungsmetriken. Sie können beispielsweise festlegen, dass AWS OpsWorks Stacks Instances starten, wenn die durchschnittliche CPU-Auslastung 80% übersteigt, und Instances beenden, wenn die durchschnittliche CPU-Last unter 60% fällt.

Für Linux-Stacks werden sowohl zeitbasierte als auch lastbasierte Instances unterstützt, während für Windows-Stacks nur zeitbasierte Instances unterstützt werden.

Im Gegensatz zu 24/7-Instances, welche manuell gestartet und angehalten werden müssen, starten Sie keine zeit- oder lastbasierten Instances selbst oder halten diese an. Stattdessen konfigurieren Sie die Instances und AWS OpsWorks Stacks startet oder stoppt sie je nach Konfiguration. Sie konfigurieren beispielsweise zeitbasierte Instances so, dass sie nach einem bestimmten Zeitplan gestartet und gestoppt werden. AWS OpsWorks Stacks startet und stoppt dann die Instanzen entsprechend dieser Konfiguration.

Üblicherweise werden alle drei Instance-Typen wie folgt gemeinsam verwendet.

- Eine Gruppe von 24/7-Instances, um die Grundlast zu verarbeiten. Sie starten diese Instances in der Regel einfach und lassen sie unterbrechungsfrei ausführen.
- Eine Reihe von zeitbasierten Instances, die AWS OpsWorks Stacks starten und stoppen, um vorhersehbare Verkehrsschwankungen zu bewältigen. Wenn Ihr Datenverkehr z. B. während der Arbeitszeit am höchsten ist, konfigurieren Sie die zeitbasierten Instances so, dass sie morgens starten und abends anhalten.
- Eine Reihe von lastbasierten Instances, die AWS OpsWorks Stacks startet und stoppt, um unvorhersehbare Verkehrsschwankungen zu bewältigen. AWS OpsWorks Stacks startet sie, wenn sich die Auslastung der Kapazität der rund um die Uhr verfügbaren und zeitbasierten Instances der Stacks nähert, und stoppt sie, wenn sich der Verkehr wieder normalisiert.

Weitere Informationen zur Verwendung dieser Skalierungszeiten finden Sie unter Optimieren der Serveranzahl.

> **Note**
>
> Wenn Sie Apps für die Ebene der Instanzen oder benutzerdefinierte Kochbücher erstellt haben, stellt AWS OpsWorks Stacks beim ersten Start automatisch die neueste Version für zeit- und lastbasierte Instanzen bereit. AWS OpsWorks Stacks stellt jedoch nicht unbedingt die neuesten Kochbücher für neu gestartete Offline-Instanzen bereit. Weitere Informationen finden Sie unter Bearbeiten von Anwendungen und Aktualisieren von benutzerdefinierten Rezeptbüchern.

Themen

- Verwendung der automatischen zeitbasierten Skalierung
- Verwenden Sie die automatische lastbasierte Skalierung
- Wie sich lastbasierte Skalierung von Auto Healing unterscheidet

Verwendung der automatischen zeitbasierten Skalierung

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Mit der zeitbasierten Skalierung können Sie steuern, wie viele Instances ein Layer zu bestimmten Tages- oder Wochentagen online haben soll, indem Sie Instances nach einem bestimmten Zeitplan starten oder stoppen. AWS OpsWorks Stacks überprüft alle paar Minuten und startet oder stoppt Instances nach Bedarf. Sie geben den Zeitplan folgendermaßen für jede Instance separat an:

- Uhrzeit. Sie können zum Beispiel mehrere Instances tagsüber ausführen als nachts.
- Wochentag. Sie können z. B. mehrere Instances an Wochentagen ausführen als an Wochenenden.

> **Note**
>
> Sie können keine bestimmten Datumsangaben angeben.

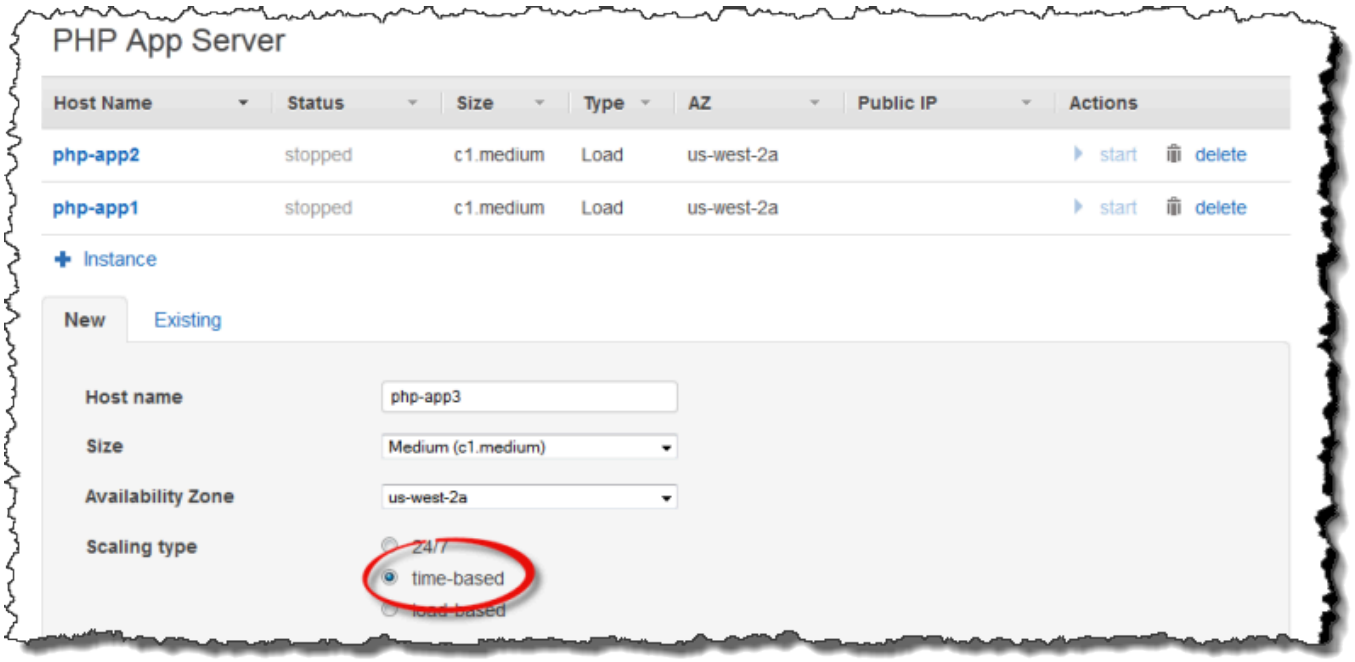
Themen

Hinzufügen einer zeitbasierten Instanz zu einer Ebene

Sie können entweder eine neue zeitbasierte Instance zu dem Layer hinzufügen oder eine bestehende Instanz verwenden.

So fügen Sie eine neue zeitbasierte Instance hinzu

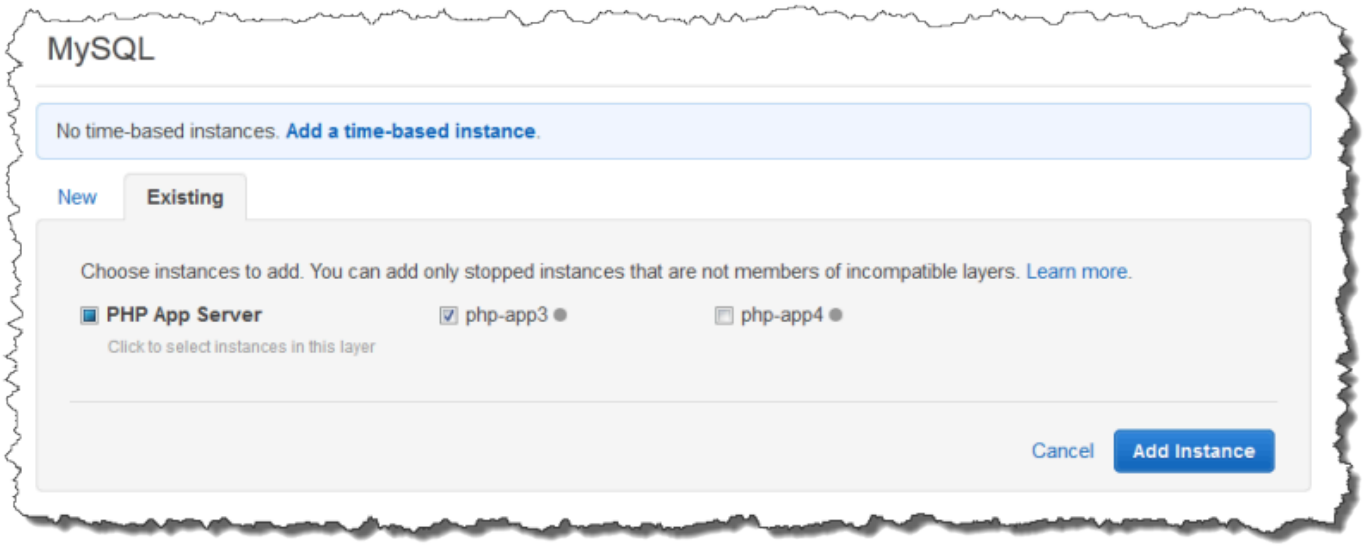
1. Wählen Sie auf der Seite Instances die Option + Instance, um eine Instanz hinzuzufügen. Wählen Sie auf der Registerkarte Neu die Option Erweitert und dann zeitbasiert aus.



2. Konfigurieren Sie die Instanz. Wählen Sie dann Add Instance, um die Instanz dem Layer hinzuzufügen.

So fügen Sie eine vorhandene zeitbasierte Instance zu einem Layer hinzu

1. Wählen Sie auf der Seite Zeitbasierte Instanzen die Option + Instanz aus, wenn ein Layer bereits über eine zeitbasierte Instanz verfügt. Wählen Sie andernfalls Eine zeitbasierte Instanz hinzufügen aus. Wählen Sie dann den Tab Existierend.



2. Wählen Sie auf der Registerkarte Existierend eine Instanz aus der Liste aus. Die Liste zeigt nur zeitbasierte Instances an.

> **Note**
>
> Wenn Sie Ihre Meinung zur Verwendung einer vorhandenen Instanz ändern, erstellen Sie auf der Registerkarte Neu eine neue Instanz, wie im vorherigen Verfahren beschrieben.

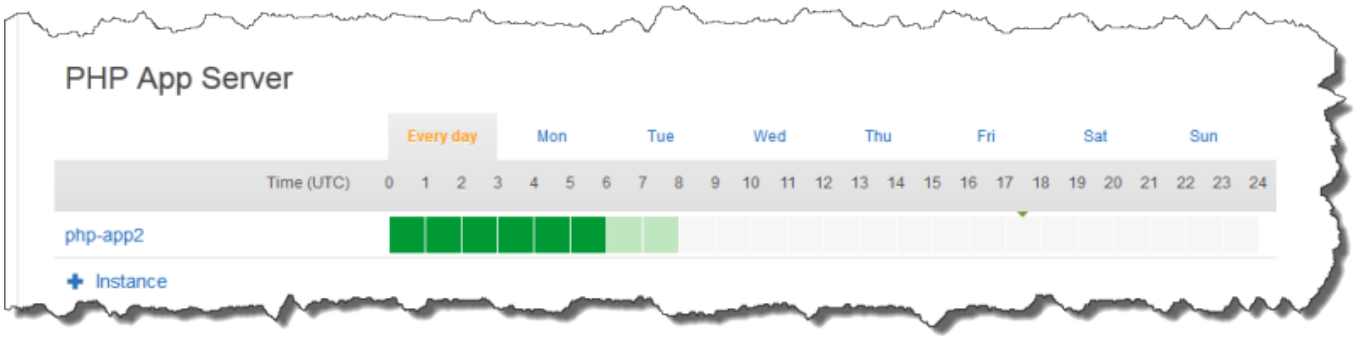
3. Wählen Sie Instanz hinzufügen, um die Instanz dem Layer hinzuzufügen.

Konfiguration einer zeitbasierten Instanz

Nachdem Sie eine zeitbasierte Instance zum Layer hinzufügen, konfigurieren Sie ihren Zeitplan wie folgt ein.

So konfigurieren Sie eine zeitbasierte Instance

1. Wählen Sie im Navigationsbereich unter Instances die Option Time-based aus.
2. Geben Sie die Online-Perioden für jede zeitbasierte Instanz an, indem Sie die entsprechenden Felder unter der gewünschten Stunde ausfüllen.
   - Um jeden Tag denselben Zeitplan zu verwenden, wählen Sie die Registerkarte Jeden Tag und geben Sie dann die Online-Zeiträume an.
   - Um an verschiedenen Tagen unterschiedliche Zeitpläne zu verwenden, wählen Sie jeden Tag und dann die entsprechenden Zeiträume aus.



> **Note**
>
> Achten Sie darauf, dass Sie die Zeit berücksichtigen, die zum Starten einer Instance benötigt wird, und dass AWS OpsWorks Stacks nur alle paar Minuten überprüft, ob Instances gestartet oder gestoppt werden sollten. Wenn eine Instance z. B. um 1:00 Uhr ausgeführt werden soll, starten Sie sie um 0:00 Uhr. Andernfalls startet AWS OpsWorks Stacks die Instance möglicherweise erst einige Minuten nach 1:00 UTC, und es dauert noch einige Minuten, bis die Instance online ist.

Sie können die Online-Zeiträume einer Instance jederzeit ändern, indem Sie die vorherigen Schritte ausführen. Bei der nächsten Überprüfung durch AWS OpsWorks Stacks wird anhand des neuen Zeitplans bestimmt, ob Instances gestartet oder gestoppt werden sollen.

> **Note**
>
> Sie können einem Layer eine neue zeitbasierte Instanz hinzufügen, indem Sie die Seite Zeitbasiert öffnen und Zeitbasierte Instanz hinzufügen (falls Sie dem Layer noch keine zeitbasierte Instanz hinzugefügt haben) oder + Instanz (wenn der Layer bereits über eine oder mehrere zeitbasierte Instanzen verfügt) auswählen. Konfigurieren Sie dann die Instanz wie in den vorherigen Verfahren beschrieben.

Verwenden Sie die automatische lastbasierte Skalierung

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Mit lastbasierten Instances können Sie Instances als Reaktion auf Änderungen des eingehenden Datenverkehrs schnell starten oder stoppen. AWS OpsWorks Stacks verwendet CloudWatchAmazon-Daten, um die folgenden Metriken für jede Ebene zu berechnen, die Durchschnittswerte für alle Instanzen der Ebene darstellen:

- CPU: Die durchschnittliche CPU-Auslastung, z. B. 80 %
- Speicher: Die durchschnittliche Speicherbelegung, z. B. 60 %
- Last: Die durchschnittliche numerische Arbeit, die ein System in einer Minute ausführt.

Sie definieren Schwellenwerte zum Hochskalieren und Herunterskalieren für einzelne oder alle diese Metriken. Sie können auch benutzerdefinierte CloudWatch Alarme als Schwellenwerte verwenden.

Das Überschreiten eines Schwellenwertes löst ein Skalierungsereignis aus. Sie bestimmen, wie AWS OpsWorks Stacks Skalierungsereignisse beantwortet, indem Sie die folgenden Schritte angeben:

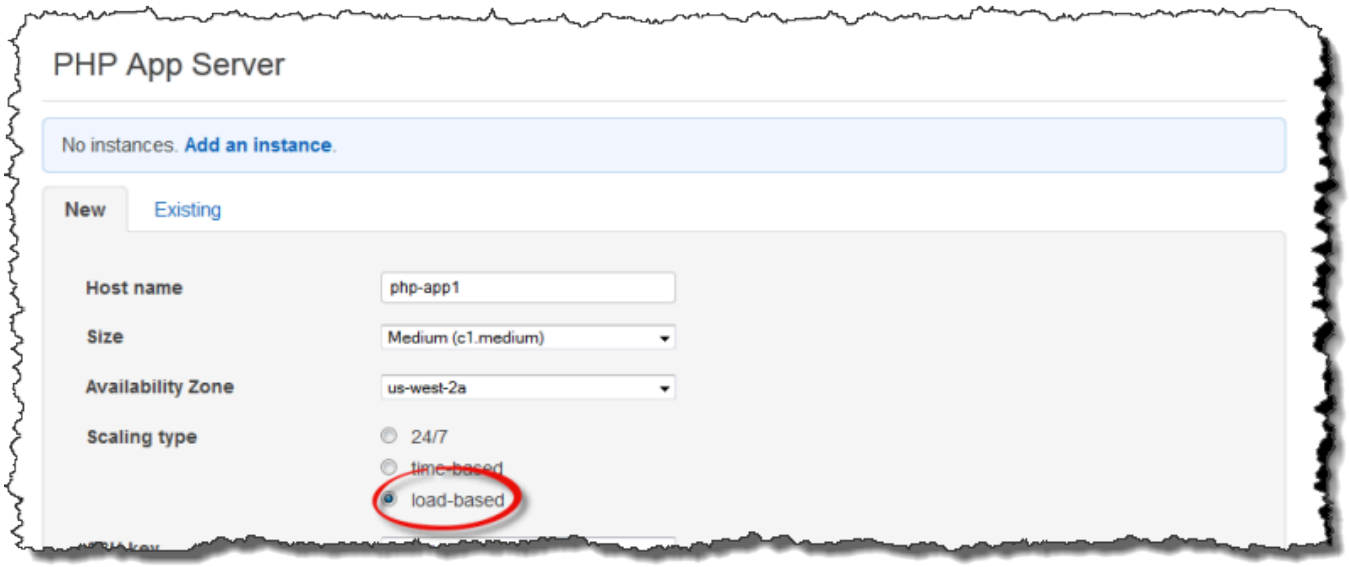
- Wie viele Instances zu starten oder anzuhalten sind.
- Wie lange AWS OpsWorks Stacks warten sollen, nachdem sie einen Schwellenwert überschritten haben, bevor sie Instances starten oder löschen. Eine CPU-Auslastung z. B. muss für mindestens 15 Minuten über dem Schwellenwert liegen. Dieser Wert erlaubt Ihnen, kurze Datenverkehrschwankungen zu ignorieren.
- Wie lange AWS OpsWorks Stacks nach dem Starten oder Stoppen von Instances warten sollten, bevor die Metriken erneut überwacht werden. In der Regel möchten Sie, dass gestartete Instances genügend Zeit haben, online zu gehen, oder angehaltene Instances genügend Zeit haben, herunterzufahren, bevor Sie bewerten, ob der Layer immer noch einen Schwellenwert überschreitet.

Wenn ein Skalierungsereignis eintritt, startet oder stoppt AWS OpsWorks Stacks nur lastbasierte Instances. Es werden keine 24/7- oder zeitbasierte Instances gestartet oder angehalten.

> **Note**
>
> Die automatische lastbasierte Skalierung erstellt keine neuen Instances. Sie startet und beendet lediglich Instances, die Sie erstellt haben. Sie müssen daher im Voraus genügend lastbasierte Instances bereitstellen, um die maximal zu erwartende Last zu verarbeiten.

So erstellen Sie eine lastbasierte Instance

1. Wählen Sie auf der Seite Instances die Option +Instance aus, um eine Instanz hinzuzufügen. Wählen Sie Advanced und anschließend Load-based aus.



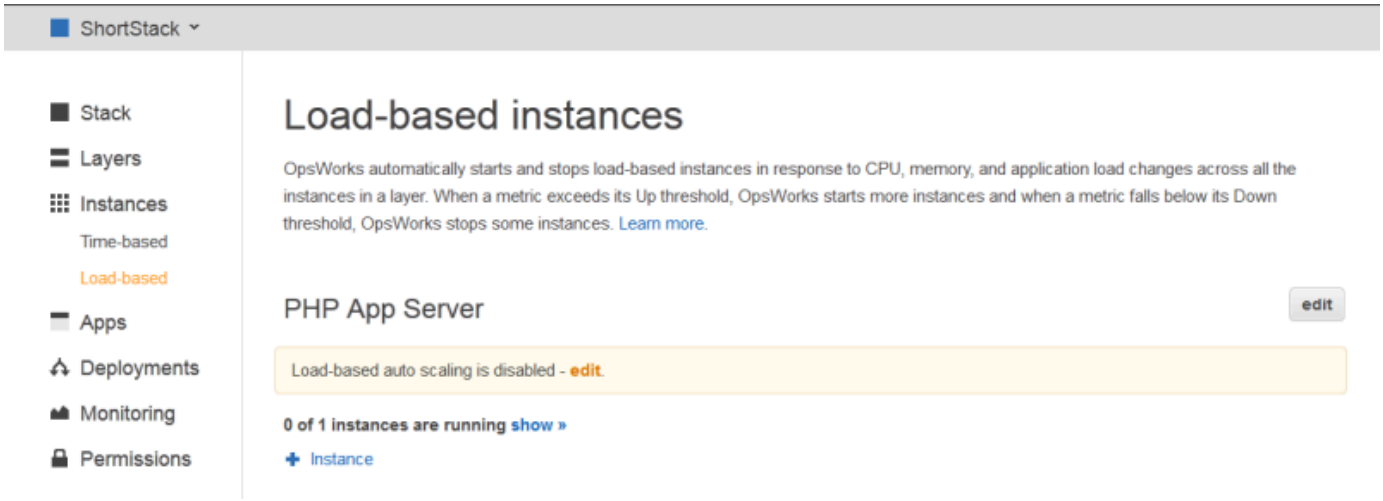
2. Konfigurieren Sie die Instanz und wählen Sie dann Instanz hinzufügen, um die Instanz dem Layer hinzuzufügen.

Wiederholen Sie dieses Verfahren, bis Sie eine ausreichende Anzahl von Instances erstellt haben. Sie können Instances zu einem späteren Zeitpunkt je nach Bedarf hinzufügen oder entfernen.

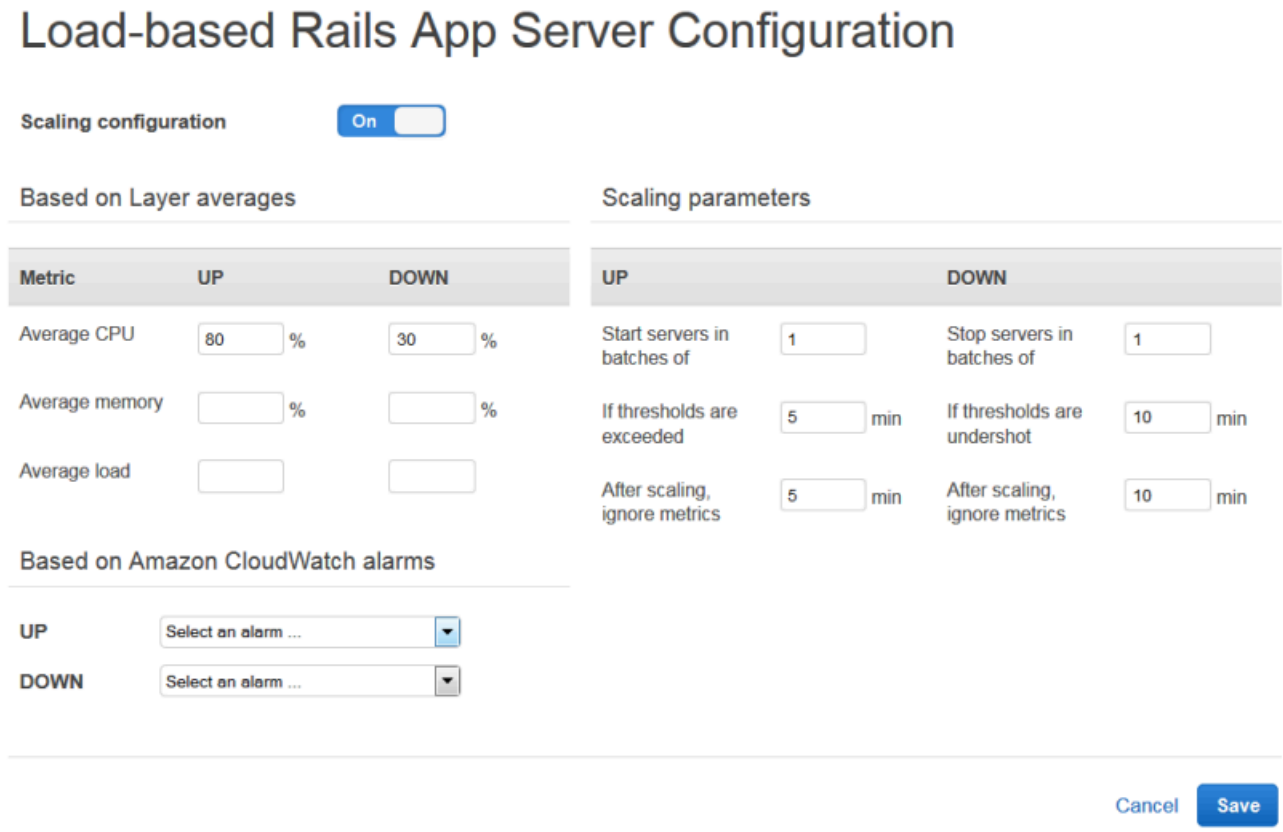
Nachdem Sie lastbasierte Instances zu einem Layer hinzugefügt haben, aktivieren Sie die lastbasierte Skalierung und geben Sie die Konfiguration an. Die Konfiguration der lastbasierten Skalierung ist eine Eigenschaft des Layers und nicht der Instance, in der angegeben wird, wann ein Layer seine lastbasierten Instances starten oder beenden sollte. Sie muss für jeden Layer separat angegeben werden, der lastbasierte Instances verwendet.

So aktivieren und konfigurieren Sie eine automatisierte lastbasierte Skalierung

1. Wählen Sie im Navigationsbereich unter Instances die Option Load-based aus und wählen Sie dann Bearbeiten für den entsprechenden Layer aus.



2. Stellen Sie Load-based Auto Scaling aktiviert auf Ein. Legen Sie dann den Schwellenwert und die Skalierungsparameter fest, um zu definieren, wie und wann Instances hinzuzufügen oder zu löschen sind.



Durchschnittliche Schwellenwerte des Layers

Sie können die Skalierung von Schwellenwerten basierend auf den folgenden Werten einstellen, die den Durchschnitt aller Instances des Layers angeben.

- Durchschnittliche CPU-Auslastung — Die durchschnittliche CPU-Auslastung des Layers als Prozentsatz der Gesamtleistung.
- Durchschnittlicher Arbeitsspeicher — Die durchschnittliche Speicherauslastung der Schicht als Prozentsatz der Gesamtspeichernutzung.
- Durchschnittliche Last — Die durchschnittliche Auslastung der Ebene.

  Weitere Informationen zur Berechnung der Last finden Sie unter Last (Berechnung) auf Wikipedia.

Das Überschreiten eines Schwellenwerts führt zu einem Skalierungsereignis, bei dem eine Hochskalierung erfolgt, wenn mehr Instances benötigt werden, und eine Herunterskalierung, wenn weniger Instances benötigt werden. AWS OpsWorks Stacks fügt dann Instanzen auf der Grundlage der Skalierungsparameter hinzu oder löscht sie.

Benutzerdefinierte Alarme CloudWatch

Sie können bis zu fünf benutzerdefinierte CloudWatch Alarme als Schwellenwerte für das Hoch- oder Herunterskalieren verwenden. Sie müssen in derselben Region wie der Stack sein. Weitere Informationen zum Erstellen benutzerdefinierter Alarme finden Sie unter CloudWatch Amazon-Alarme erstellen.

> **Note**
>
> Zur Verwendung benutzerdefinierter Alarme müssen Sie Ihre Service-Rolle aktualisieren, um `cloudwatch:DescribeAlarms` zu erlauben. Sie können AWS OpsWorks Stacks entweder die Rolle für Sie aktualisieren lassen, wenn Sie diese Funktion zum ersten Mal verwenden, oder Sie können die Rolle manuell bearbeiten. Weitere Informationen finden Sie unter AWS OpsWorks Stacks erlauben, in Ihrem Namen zu handeln.
>
> Wenn mehrere Alarme für die lastbasierte Konfiguration konfiguriert sind und sich ein Alarm im `INSUFFICIENT_DATA` metrischen Alarmstatus befindet, kann die lastbasierte Instanzskalierung nicht erfolgen, selbst wenn sich ein anderer Alarm im Status befindet. `ALARM` Die automatische Skalierung kann nur fortgesetzt werden, wenn sich alle Alarme im Status `OK` oder `ALARM` befinden. Weitere Informationen zur Verwendung von CloudWatch Amazon-Alarmen finden Sie unter CloudWatch Amazon-Alarme verwenden im CloudWatch Amazon-Benutzerhandbuch.

Skalierungsparameter

Die folgenden Parameter steuern, wie AWS OpsWorks Stacks Skalierungsereignisse verwaltet.

- Server stapelweise starten — Die Anzahl der Instanzen, die hinzugefügt oder entfernt werden sollen, wenn das Skalierungsereignis eintritt.
- Wenn Schwellenwerte überschritten werden — Der Zeitraum (in Minuten), in dem die Last über einem Upscaling-Schwellenwert oder unter einem Downscaling-Schwellenwert bleiben muss, bevor AWS OpsWorks Stacks ein Skalierungsereignis auslöst.
- Metriken nach der Skalierung ignorieren — Der Zeitraum (in Minuten), in dem AWS OpsWorks Stacks Metriken ignorieren und zusätzliche Skalierungsereignisse unterdrücken soll, nachdem ein Skalierungsereignis eingetreten ist.

  AWS OpsWorks Stacks fügt beispielsweise nach einem Upscaling-Ereignis neue Instances hinzu, aber die Instances beginnen erst, die Last zu reduzieren, wenn sie gebootet und konfiguriert wurden. Es macht keinen Sinn, weitere Skalierungsereignisse auszulösen, bis die neuen Instances online sind, und Anfragen zu bearbeiten, was in der Regel ein paar Minuten dauert. Diese Einstellung ermöglicht Ihnen, AWS OpsWorks Stacks anzuweisen, Skalierungsereignisse lange genug zu unterdrücken, um die neuen Instances online zu stellen.

  Sie können diese Einstellung erhöhen, um plötzliche Skalierungsschwankungen zu verhindern, wenn Layer-Durchschnittswerte wie Durchschnittliche CPU, Durchschnittlicher Arbeitsspeicher oder Durchschnittliche Auslastung vorübergehend nicht übereinstimmen.

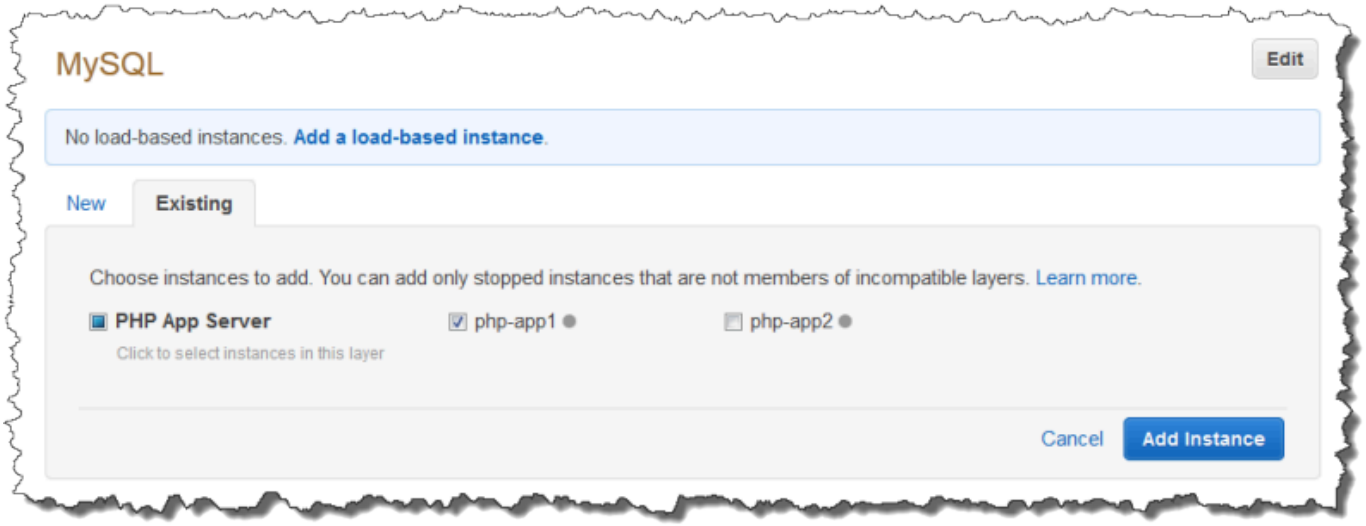
  Wenn die CPU-Auslastung beispielsweise über dem Grenzwert liegt und die Speicherauslastung kurz vor dem Herunterskalieren steht, kann auf ein Instance-Upscale-Ereignis unmittelbar ein Ereignis zum Herunterskalieren des Speichers folgen. Um dies zu verhindern, können Sie die Anzahl der Minuten in der Einstellung Metriken ignorieren nach der Skalierung erhöhen. In diesem Beispiel würde die CPU-Skalierung stattfinden, das Ereignis zum Herunterskalieren des Speichers jedoch nicht.

3. Um weitere lastbasierte Instances hinzuzufügen, wählen Sie + Instance, konfigurieren Sie die Einstellungen und wählen Sie dann Add Instance aus. Wiederholen Sie den Vorgang, bis Sie genügend lastbasierte Instances haben, um die maximal zu erwartetende Last zu verarbeiten. Wählen Sie dann Speichern.

> **Note**
>
> Sie können einem Layer auch eine neue lastbasierte Instanz hinzufügen, indem Sie die Seite Lastenbasiert öffnen und eine lastbasierte Instanz hinzufügen (falls Sie dem Layer noch keine lastbasierte Instanz hinzugefügt haben) oder + Instanz (wenn der Layer bereits über eine oder mehrere lastbasierte Instanzen verfügt) auswählen. Anschließend konfigurieren Sie die Instance wie weiter oben beschrieben.

So fügen Sie eine vorhandene lastbasierte Instance zu einem Layer hinzu

1. Wählen Sie im Navigationsbereich unter Instances die Option Load-based aus.
2. Wenn Sie die lastbasierte automatische Skalierung für einen Layer bereits aktiviert haben, wählen Sie + Instanz aus. Andernfalls wählen Sie Eine lastbasierte Instanz hinzufügen. Wählen Sie die Registerkarte „Bestehend“.



3. Wählen Sie auf der Registerkarte Existierend eine Instanz aus. Die Liste zeigt nur lastbasierte Instances.

> **Note**
>
> Wenn Sie Ihre Meinung zur Verwendung einer vorhandenen Instanz ändern, erstellen Sie auf der Registerkarte Neu eine neue Instanz, wie im vorherigen Verfahren beschrieben.

4. Wählen Sie „Instanz hinzufügen“, um die Instanz dem Layer hinzuzufügen.

Sie können die Konfiguration für die automatische lastbasierte Skalierung jederzeit bearbeiten oder diese Skalierung deaktivieren.

So deaktivieren Sie eine automatische lastbasierte Skalierung

1. Wählen Sie im Navigationsbereich unter Instances die Option Load-based und anschließend für den entsprechenden Layer Bearbeiten aus.
2. Schalten Sie Load-based Auto Scaling aktiviert auf Nein.

Wie sich lastbasierte Skalierung von Auto Healing unterscheidet

Die automatische lastbasierte Skalierung verwendet Metriken, die den Durchschnitt aller ausgeführten Instances ermitteln. Wenn die Metriken zwischen den angegebenen Schwellenwerten bleiben, startet oder stoppt AWS OpsWorks Stacks keine Instances. Bei der auto Heilung hingegen startet AWS OpsWorks Stacks automatisch eine neue Instanz mit derselben Konfiguration, wenn eine Instanz nicht mehr reagiert. Die Instance kann voraussichtlich nicht reagieren, da ein Netzwerkproblem oder ein Problem mit der Instance besteht.

Nehmen wir zum Beispiel an, Ihr CPU-Upscaling-Schwellenwert liegt bei 80% und eine Instance reagiert nicht mehr.

- Wenn die auto Heilung deaktiviert ist und die verbleibenden laufenden Instances die durchschnittliche CPU-Auslastung unter 80% halten können, startet AWS OpsWorks Stacks keine neue Instanz. Es startet eine Ersatz-Instance nur dann, wenn die durchschnittliche CPU-Auslastung über die verbleibenden Instances 80 % übersteigt.
- Wenn Auto Healing aktiviert ist, startet AWS OpsWorks Stacks unabhängig von den Lastschwellenwerten eine Ersatzinstanz.

Verwenden von Computing-Ressourcen, die nicht mit AWS OpsWorks Stacks erstellt wurden

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Funktion wird nur für Linux-Stacks unterstützt.

Instancesbeschreibt, wie AWS OpsWorks Stacks verwendet werden, um Gruppen von Amazon Elastic Compute Cloud (Amazon EC2) -Instances zu erstellen und zu verwalten. Sie können Linux-Rechenressourcen auch in einen Stack integrieren, der außerhalb von AWS OpsWorks Stacks erstellt wurde:

- Amazon EC2 EC2-Instances, die Sie direkt mit der Amazon EC2 EC2-Konsole, CLI oder API erstellt haben.
- Lokale Instances, die auf Ihrer eigenen Hardware ausgeführt werden, einschließlich Instances auf virtuellen Maschinen.

Diese Rechenressourcen werden zu von AWS OpsWorks Stacks verwalteten Instances, und Sie können sie ähnlich wie normale Stacks-Instances verwalten: AWS OpsWorks

- Benutzerberechtigungen verwalten — Mithilfe der AWS OpsWorks Stacks-Benutzerverwaltung können Sie angeben, welche Benutzer auf Ihre Stacks zugreifen dürfen, welche Aktionen sie auf den Instanzen des Stacks ausführen dürfen und ob sie über SSH-Zugriff und Sudo-Rechte verfügen.
- Automatisieren Sie Aufgaben — Sie können AWS OpsWorks Stacks benutzerdefinierte Chef-Rezepte ausführen lassen, um Aufgaben wie das Ausführen von Skripten auf einer oder allen Instanzen eines Stacks mit einem einzigen Befehl auszuführen.

  Wenn Sie die Instanz einer Ebene zuweisen, führt AWS OpsWorks Stacks an wichtigen Punkten ihres Lebenszyklus automatisch einen bestimmten Satz von Chef-Rezepten auf der Instanz aus, einschließlich Ihrer benutzerdefinierten Rezepte. Beachten Sie, dass Sie registrierte Amazon EC2 EC2-Instances nur benutzerdefinierten Layern zuweisen können.
- Ressourcen verwalten — Mit einem Stack können Sie Ressourcen in einem Stack gruppieren und verwalten AWS-Region, und das OpsWorks Dashboard zeigt den Status Ihrer Stacks in allen Regionen an.

- Pakete installieren — Sie können Chef-Rezepte verwenden, um Pakete auf einer beliebigen Instanz in einem Stack zu installieren.
- Aktualisieren Sie das Betriebssystem — AWS OpsWorks Stacks bietet eine einfache Möglichkeit, Betriebssystem-Sicherheitspatches und -updates auf den Instanzen eines Stacks zu installieren.
- Anwendungen bereitstellen — AWS OpsWorks Stacks stellt Anwendungen konsistent auf allen Anwendungsserverinstanzen des Stacks bereit.
- Überwachung — AWS OpsWorks Stacks erstellt benutzerdefinierte CloudWatchMetriken zur Überwachung aller Instanzen Ihres Stacks.

Preisinformationen finden Sie unter OpsWorks AWS-Preise.

Nachfolgend finden Sie das grundlegende Verfahren zum Verwenden einer registrierten Instance.

1. Registrieren Sie die Instance für einen Stack.

   Die Instance ist jetzt Teil des Stacks und wird von AWS OpsWorks Stacks verwaltet.
2. Optional können Sie die Instance einem Layer zuweisen.

   In diesem Schritt können Sie die Verwaltungsfunktionen von AWS OpsWorks Stacks in vollem Umfang nutzen. Sie können jeder Ebene registrierte lokale Instances zuweisen. Registrierte Amazon EC2 EC2-Instances können nur benutzerdefinierten Layern zugewiesen werden.
3. Verwenden Sie AWS OpsWorks Stacks, um die Instance zu verwalten.
4. Wenn Sie die Instance im Stack nicht mehr benötigen, melden Sie sie ab. Dadurch wird die Instance aus Stacks entfernt. AWS OpsWorks

In den folgenden Abschnitten wird dieses Verfahren ausführlich beschrieben.

Themen

Registrierung einer Instance bei einem Stacks-Stack AWS OpsWorks

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Diese Funktion wird nur für Linux-Stacks unterstützt.

Um eine Instanz zu registrieren, die sich außerhalb von AWS OpsWorks Stacks befindet, führen Sie den Befehl aus. AWS CLI aws opsworks register Diesen Befehl können Sie auf der Instance, die Sie registrieren möchten, oder von einem anderen Computer aus ausführen. Sie wenden die `AWSOpsWorksRegisterCLI_OnPremises` Richtlinien `AWSOpsWorksRegisterCLI_EC2` oder auf einen Benutzer oder eine Gruppe an, um Berechtigungen zu erteilen, die für die AWS CLI Registrierung von EC2- bzw. lokalen Instances erforderlich sind. Für diese Richtlinien ist Version 1.16.180 oder neuer erforderlich. AWS CLI

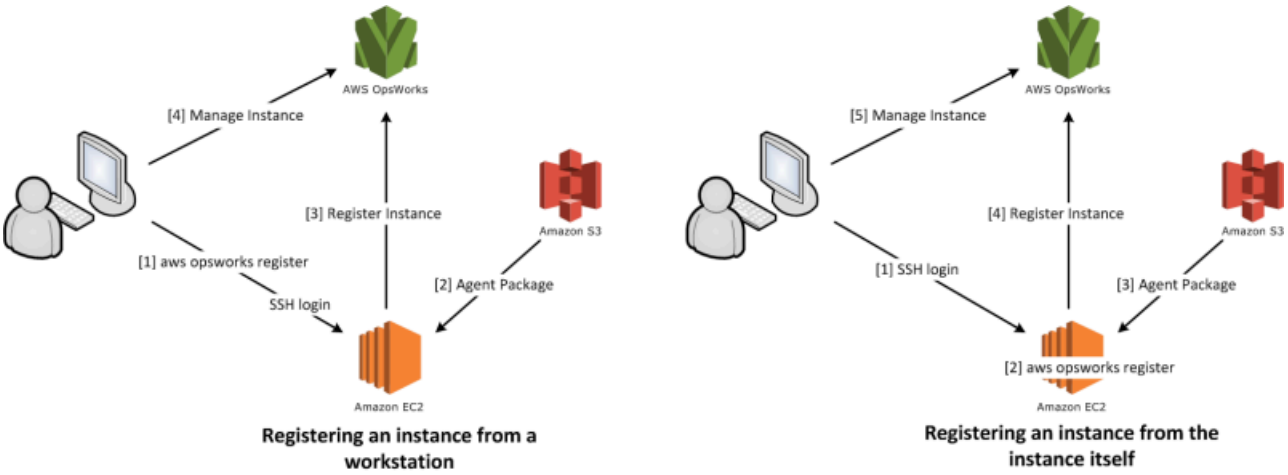
Note

Um zu verhindern, dass Benutzer oder Rollen Instanzen registrieren, aktualisieren Sie das Instanzprofil, um den Zugriff auf den register Befehl zu verweigern.

Bei der Registrierung wird ein Agent auf einer Instance installiert, die Sie mithilfe von AWS OpsWorks Stacks verwalten möchten, und die Instance wird mit einem von Ihnen angegebenen AWS OpsWorks Stack registriert. Eine Instance wird nach ihrer Registrierung Teil des Stacks und von AWS OpsWorks Stacks verwaltet. Weitere Informationen finden Sie unter Verwalten von registrierten Instances.

> **Note**
>
> AWS Tools for PowerShell enthält zwar das `Register-OpsInstance`Cmdlet, das die `register` API-Aktion aufruft, wir empfehlen jedoch, den `register` Befehl stattdessen AWS CLI mit dem auszuführen.

Das folgende Diagramm zeigt beide Ansätze zur Registrierung einer Amazon EC2 EC2-Instance. Mit denselben Methoden können Sie eine lokale Instance registrieren.



> **Note**
>
> Mit der AWS OpsWorks Stacks-Konsole können Sie eine registrierte Instance verwalten, aber für die Instance-Registrierung selbst müssen Sie den AWS CLI-Befehl `register` verwenden. Der Grund dafür ist, dass die Registrierung auf der Instance erfolgen muss – und das ist über die Konsole nicht möglich.

In den folgenden Abschnitten wird dieses Verfahren ausführlich beschrieben.

Themen

- Anleitung: Registrieren einer Instance von der Workstation
- Registrierung von Amazon EC2- und lokalen Instances

Anleitung: Registrieren einer Instance von der Workstation

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Funktion wird nur für Linux-Stacks unterstützt.

Der Registrierungsprozess unterstützt mehrere Szenarien. Dieser Abschnitt führt Sie durch ein end-to-end Beispiel für ein Szenario: wie Sie Ihre Workstation verwenden, um eine Amazon EC2 EC2-Instance zu registrieren. In den anderen Registrierungsszenarien wird eine vergleichbare Methode verwendet. Weitere Informationen finden Sie unter Registrierung von Amazon EC2- und lokalen Instances.

> **Note**
>
> In der Regel möchten Sie eine bestehende Amazon EC2 EC2-Instance registrieren. Sie können aber einfach eine neue Instance und einen neuen Stack für diese Anleitung erstellen und wieder löschen, wenn Sie fertig sind.

Themen

Schritt 1: Erstellen von Stack und Instance

Zu Beginn benötigen Sie einen Stack und eine Amazon EC2 EC2-Instance, die bei diesem Stack registriert sein müssen.

So erstellen Sie den Stack und die Instance

1. Erstellen Sie in der AWS OpsWorks Stacks-Konsole einen neuen Stack mit der Bezeichnung **`EC2Register`**. Übernehmen Sie die Standardwerte für die anderen Stack-Einstellungen.
2. Starten Sie eine neue Instance von der Amazon EC2 EC2-Konsole aus. Beachten Sie Folgendes:

   - Die Instance muss in derselben Region und VPC sein wie der Stack.

     Falls Sie eine VPC nutzen, wählen Sie ein öffentliches Subnetz für diese Anleitung aus.
   - Sofern Sie einen SSH-Schlüssel erstellen müssen, speichern Sie die Datei mit dem privaten Schlüssel auf der Workstation und notieren Sie den Namen und den Speicherort der Datei.

     Wenn Sie einen vorhandenen Schlüssel verwenden, notieren Sie den Namen und den Speicherort der Datei mit dem privaten Schlüssel. Diese Daten benötigen Sie später.
   - Die Instance muss auf einem der unterstützten Linux-Betriebssysteme basieren. Wenn sich Ihr Stack beispielsweise in USA West (Oregon) befindet, können Sie `ami-35501205` damit eine Ubuntu 14.04 LTS-Instance in dieser Region starten.

   Übernehmen Sie andernfalls die Standardwerte.

Während die Instance gestartet wird, können Sie mit dem nächsten Abschnitt fortfahren.

Schritt 2: Installieren und Konfigurieren der AWS CLI

Die Registrierung erfolgt mit dem Befehl. AWS CLI aws opsworks register Bevor Sie Ihre erste Instanz registrieren, müssen Sie Version 1.16.180 von AWS CLI oder neuer ausführen. Die Installationsdetails hängen vom Betriebssystem Ihrer Workstation ab. Weitere Informationen zur Installation von finden Sie unter Installation der AWS-Befehlszeilenschnittstelle. AWS CLI Um zu überprüfen, welche Version der AWS CLI Sie ausführen, geben Sie in einer Shell-Sitzung `aws --version` ein.

> **Note**
>
> Um zu verhindern, dass Benutzer oder Rollen Instances registrieren, aktualisieren Sie das Instance-Profil, um den Zugriff auf den register Befehl zu verweigern.

Wir empfehlen dringend, diesen Schritt nicht zu überspringen, auch wenn Sie den bereits AWS CLI auf Ihrer Workstation ausführen. Die Verwendung der neuesten Veröffentlichung der AWS CLI stellt eine bewährte Sicherheitsmethode dar.

Sie müssen `register` mit einer Reihe von AWS-Anmeldeinformationen, die über entsprechende Berechtigungen verfügen, bereitstellen. Um zu vermeiden, dass Anmeldeinformationen direkt auf einer Instance installiert werden, wird empfohlen, Instances zu registrieren, die mit einem Instanzprofil gestartet werden, und dann den `--use-instance-profile` Switch zu Ihrem Befehl hinzuzufügen. `register` Wenn Sie Anmeldeinformationen aus einem Instance-Profil erhalten, gehen Sie weiter zu Schritt 3: Registrieren der Instance für das EC2Register-Stack in diesem Thema. Wenn Ihre Instance jedoch nicht mit einem Instance-Profil gestartet wurde, können Sie einen IAM-Benutzer erstellen. Das folgende Verfahren erstellt einen neuen Benutzer mit den entsprechenden Berechtigungen, installiert die Anmeldeinformationen des Benutzers auf der Workstation und gibt diese Anmeldeinformationen dann an weiter. `register`

> **Warning**
>
> IAM-Benutzer verfügen über langfristige Anmeldeinformationen, was ein Sicherheitsrisiko darstellt. Um dieses Risiko zu minimieren, empfehlen wir, diesen Benutzern nur die Berechtigungen zu gewähren, die sie für die Ausführung der Aufgabe benötigen, und diese Benutzer zu entfernen, wenn sie nicht mehr benötigt werden.

So erstellen Sie den Benutzer

1. Wählen Sie in der IAM-Konsole im Navigationsbereich die Option Users und dann Add user aus.
2. Fügen Sie einen Benutzer mit dem Namen **`EC2Register`** hinzu.
3. Wählen Sie Weiter aus.
4. Wählen Sie auf der Seite Berechtigungen festlegen die Option Richtlinien direkt anhängen aus.
5. Geben Sie **`OpsWorks`** in das Filterfeld „Berechtigungsrichtlinie“ ein, um die AWS OpsWorks Richtlinien anzuzeigen, wählen Sie eine der folgenden Richtlinien aus und klicken Sie dann auf

Weiter: Überprüfung. Diese Richtlinie gewährt dem Benutzer die erforderlichen Berechtigungen zur Ausführung von `register`.

- Wählen Sie `AWSOpsWorksRegisterCLI_EC2` aus, um den Benutzer zum Registrieren von EC2-Instances, die Instance-Profile verwenden, zu berechtigen.
- Wählen Sie `AWSOpsWorksRegisterCLI_OnPremises` aus, um den Benutzer zum Registrieren von lokalen Instances zu berechtigen.

6. Wählen Sie Weiter aus.
7. Wählen Sie auf der Seite Review die Option Create user aus.
8. Erstellen Sie nun Zugangsschlüssel für Ihren Benutzer. Wählen Sie im Navigationsbereich Benutzer und dann den Benutzer aus, für den Sie Zugriffsschlüssel erstellen möchten.
9. Wählen Sie die Registerkarte Sicherheitsanmeldedaten und anschließend Zugriffsschlüssel erstellen aus.
10. Wählen Sie die bewährten Methoden und Alternativen für den Zugriffsschlüssel aus, die Ihrer Aufgabe am besten entsprechen.
11. Wählen Sie Weiter aus.
12. Geben Sie (optional) ein Tag ein, um die Zugriffstasten zu identifizieren.
13. Wählen Sie Weiter aus.
14. Wählen Sie „.csv-Datei herunterladen“, speichern Sie die Anmeldeinformationsdatei an einem geeigneten Ort auf Ihrem System und wählen Sie „Fertig“.

Sie müssen die Anmeldeinformationen des IAM-Benutzers für `register` bereitstellen. In dieser Anleitung werden dazu die EC2Register-Anmeldeinformationen in der `credentials`-Datei der Workstation installiert. Informationen zu anderen Methoden zur Verwaltung der AWS CLI Anmeldeinformationen für finden Sie unter Konfiguration und Anmeldeinformationsdateien.

So installieren Sie die Anmeldeinformationen des Benutzers

1. Öffnen Sie die `credentials`-Datei der Workstation oder erstellen Sie eine solche Datei. Die Datei finden Sie unter `~/.aws/credentials` (Linux, Unix und OS X) oder `C:\Users\`*`User_Name`*`\.aws\credentials` (Windows-Systeme).
2. Fügen Sie in der `credentials`-Datei ein Profil für den EC2Register-Benutzer in folgendem Format hinzu.

```
[ec2register]
aws_access_key_id = access_key_id
aws_secret_access_key = secret_access_key
```

Ersetzen Sie *access_key_id* und *secret_access_key* durch die EC2Register-Schlüssel, die Sie zuvor heruntergeladen haben.

Schritt 3: Registrieren der Instance für das EC2Register-Stack

Nun kann die Instance registriert werden.

So registrieren Sie die Instance

1. Kehren Sie AWS OpsWorks unter Stacks zum EC2Register-Stack zurück, wählen Sie im Navigationsbereich Instances und dann Instance registrieren aus.
2. Wählen Sie EC2 Instances (EC2-Instances) aus, klicken Sie auf Next: Select Instances (Weiter: Instance auswählen) und wählen Sie Ihre Instance aus der Liste aus.
3. Wählen Sie Weiter: AWS CLI installieren und Weiter: Instances registrieren. AWS OpsWorks Stacks verwendet automatisch die verfügbaren Informationen, wie die Stack-ID und die Instance-ID, um eine `register` Befehlsvorlage zu erstellen, die auf der Seite Instances registrieren angezeigt wird. In diesem Beispiel sollen Sie sich `register` mit einem SSH-Schlüssel an der Instance anmelden und dabei die Schlüsseldatei explizit angeben. Legen Sie daher I use SSH keys to connect to my instances (Ich verwende SSH-Schlüssel für die Verbindung meiner Instances) auf Yes (Ja) fest. Die Befehlsvorlage sieht etwa wie folgt aus:

```
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**
>
> Sie müssen die Region auf die Endpunktregion des AWS OpsWorks Stacks-Dienstes festlegen, nicht auf die Region des Stacks, wenn sich der Stack in einer klassischen Region befindet, die dem `us-east-1` regionalen Endpunkt zugeordnet ist. AWS OpsWorks Stacks bestimmt die Region des Stacks anhand der Stack-ID.

4. Die Befehlsvorlage enthält mehrere benutzerspezifische Argumentwerte, die durch Klammern kenntlich gemacht werden und durch entsprechende Werte zu ersetzen sind. Kopieren Sie die Befehlsvorlage in einen Texteditor und nehmen Sie die folgenden Änderungen vor.

   > **Important**
   >
   > Der IAM-Benutzer, der während des Registrierungsprozesses erstellt wird, ist während der gesamten Lebensdauer einer registrierten Instance erforderlich. Das Löschen des Benutzers führt dazu, dass der AWS OpsWorks Stacks-Agent nicht mit dem Dienst kommunizieren kann. Um Probleme bei der Verwaltung registrierter Instanzen zu vermeiden, falls der Benutzer versehentlich gelöscht wird, fügen Sie Ihrem `register` Befehl den `--use-instance-profile` Parameter hinzu, um stattdessen das integrierte Instanzprofil der Instanz zu verwenden. Durch das Hinzufügen des `--use-instance-profile` Parameters wird außerdem verhindert, dass Fehler auftreten, wenn Sie die AWS Kontozugriffsschlüssel alle 90 Tage wechseln (eine empfohlene bewährte Methode), da auf diese Weise Diskrepanzen zwischen den für den AWS OpsWorks Agenten verfügbaren Zugriffsschlüsseln und den erforderlichen IAM-Benutzern vermieden werden.

   - Ersetzen Sie die *`Schlüsseldatei`* durch den vollqualifizierten Pfad der privaten Schlüsseldatei für das Amazon EC2 EC2-Schlüsselpaar, das Sie bei der Erstellung der Instance gespeichert haben.

     Sie können ggf. einen relativen Pfad verwenden.
   - Ersetzen Sie *`username`* durch den Instance-Benutzernamen.

     In diesem Beispiel lautet der Benutzername entweder `ubuntu` (für eine Ubuntu-Instance) oder `ec2-user` (für eine Red Hat Enterprise Linux (RHEL)- oder Amazon Linux-Instance).
   - Add`--use-instance-profile`, das `register` zusammen mit dem Instance-Profil ausgeführt wird, um Fehler bei der Schlüsselrotation oder bei versehentlichem Löschen des Haupt-IAM-Benutzers zu verhindern.

   Ihr Befehl sollte in etwa wie folgt aussehen:

```
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. Öffnen Sie ein Terminalfenster auf Ihrer Workstation, fügen Sie den Befehl "`register`" aus dem Editor ein und führen Sie den Befehl aus.

   Die Registrierung dauert in der Regel etwa fünf Minuten. Wenn der Vorgang abgeschlossen ist, kehren Sie zur AWS OpsWorks Stacks-Konsole zurück und wählen Sie Fertig. Wählen Sie dann im Navigationsbereich Instances aus. Ihre Instance sollte unter Unassigned Instances aufgelistet sein. Sie können nun die Instance einem Layer zuweisen oder sie unverändert belassen. Das hängt davon ab, wie Sie die Instance verwalten möchten.
6. Wenn Sie fertig sind, beenden Sie die Instanz und löschen Sie sie dann mithilfe der AWS OpsWorks Stacks-Konsole oder mithilfe von Befehlen. Dadurch wird die Amazon EC2 EC2-Instance beendet, sodass Ihnen keine weiteren Kosten entstehen.

Registrierung von Amazon EC2- und lokalen Instances

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Funktion wird nur für Linux-Stacks unterstützt.

In diesem Abschnitt wird beschrieben, wie Sie eine Amazon EC2- oder lokale Instance bei einem AWS OpsWorks Stacks-Stack registrieren.

Themen

- Vorbereiten der Instance
- Installieren und Konfigurieren der AWS CLI

- Registrieren der Instance
- Verwenden des Befehls register
- "register"-Befehlsbeispiele
- Instance-Registrierungsrichtlinien

Vorbereiten der Instance

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Funktion wird nur für Linux-Stacks unterstützt.

Bevor Sie eine Instance registrieren können, muss die Kompatibilität mit AWS OpsWorks Stacks sichergestellt werden. Die Details hängen davon ab, ob Sie eine lokale oder eine Amazon EC2 EC2-Instance registrieren.

Lokale Instances

Eine lokale Instance muss die folgenden Kriterien erfüllen:

- Die Instance muss auf einem der unterstützten Linux-Betriebssysteme ausgeführt werden. Es ist zwar möglich, Instances mit anderen Betriebssystemen (z. B. CentOS 6.x) zu erstellen oder zu registrieren, die mithilfe von benutzerdefinierten oder in der Community generierten AMIs erstellt wurden, offiziell werden diese Betriebssysteme jedoch nicht unterstützt.

  Sie müssen das `libyaml`-Paket für die Instance installieren. Bei Ubuntu-Instances heißt das Paket `libyaml-0-2`. Bei CentOS- und Red Hat Enterprise Linux-Instances lautet der Paketname `libyaml`.

- Die Instance muss über einen unterstützten Instance-Typ verfügen (manchmal als Instance-Größe bezeichnet). Unterstützte Instance-Typen können je nach Betriebssystem variieren und hängen davon ab, ob die Stacks in einer VPC sind. Eine Liste der unterstützten Instance-Typen finden Sie in der Dropdownliste Größe, die in der AWS OpsWorks Stacks-Konsole angezeigt werden, wenn Sie versuchen, eine neue Instance in Ihrem Ziel-Stack zu erstellen. Ist ein Instance-Typ ausgegraut und kann im Ziel-Stack nicht erstellt werden, können Sie auch keine Instance dieses Typs registrieren.
- Die Instanz muss über einen Internetzugang verfügen, der es ihr ermöglicht, mit dem AWS OpsWorks Stacks-Dienstendpunkt zu kommunizieren,. `opsworks.us-east-1.amazonaws.com (HTTPS)` Die Instance muss auch ausgehende Verbindungen zu AWS-Ressourcen wie Amazon S3 unterstützen.
- Falls Sie eine Instance von einer anderen Workstation registrieren möchten, muss die registrierte Instance die SSH-Anmeldung von der Workstation unterstützen.

  Eine SSH-Anmeldung ist nicht erforderlich, wenn Sie den Registrierungsbefehl auf der Instance ausführen.
- Der AWS Zugriffsschlüssel wird für die Authentifizierung zwischen dem AWS OpsWorks Agenten und dem AWS OpsWorks Stacks-Service verwendet. Wenn Sie die Zugriffsschlüssel wie empfohlen alle 90 Tage rotieren, aktualisieren Sie den AWS OpsWorks Agenten manuell, sodass er den neuen Schlüssel verwendet. Bearbeiten Sie die `/etc/aws/opsworks/instance-agent.yml` Datei auf einem lokalen Computer oder einer lokalen Instanz mit dem neuen Zugriffsschlüssel und dem geheimen Schlüssel. Der folgende Befehl zeigt den Zugriffsschlüssel und den geheimen Schlüssel in dieser Datei. Ein Agent, der alte Schlüssel verwendet, kann Fehler verursachen.

  ```
  cat /etc/aws/opsworks/instance-agent.yml | egrep "access_key|secret_key"
  :access_key_id: AKIAIOSFODNN7EXAMPLE
  :secret_access_key: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
  ```

Amazon EC2 EC2-Instances

Eine Amazon EC2 EC2-Instance muss die folgenden Kriterien erfüllen:

- Das AMI muss auf einem der unterstützten Linux-Betriebssysteme basieren. Eine aktuelle Liste finden Sie unter AWS OpsWorks Stacks-Betriebssysteme.

  Weitere Informationen finden Sie unter Verwenden von benutzerdefinierten AMIs.

Sofern die Instance auf einem benutzerdefinierten AMI basiert, das aus einer standardmäßig unterstützten AMI abgeleitet wurde, oder sofern die Instance nur eine minimale Einrichtung aufweist, müssen Sie das `libyaml`-Paket für die Instance installieren. Bei Ubuntu-Instances heißt das Paket `libyaml-0-2`. Für Amazon Linux- und Red Hat Enterprise Linux-Instances trägt das Paket einen Namen`libyaml`.

- Die Instance muss über einen unterstützten Instance-Typ verfügen (manchmal als Instance-Größe bezeichnet). Unterstützte Instance-Typen können je nach Betriebssystem variieren und hängen davon ab, ob die Stacks in einer VPC sind. Eine Liste der unterstützten Instance-Typen finden Sie in der Dropdownliste Größe, die in der AWS OpsWorks Stacks-Konsole angezeigt werden, wenn Sie versuchen, eine neue Instance in Ihrem Ziel-Stack zu erstellen. Ist ein Instance-Typ ausgegraut und kann im Ziel-Stack nicht erstellt werden, können Sie auch keine Instance dieses Typs registrieren.
- Die Instance muss sich im Status `running` befinden.
- Die Instance darf keiner Auto Scaling-Gruppe angehören.

  Weitere Informationen finden Sie unter Detach EC2 Instances From Your Auto Scaling Group.
- Die Instance kann Teil einer VPC sein, sie muss sich jedoch in derselben VPC wie der Stack befinden und die VPC muss so konfiguriert sein, dass sie ordnungsgemäß mit Stacks funktioniert. AWS OpsWorks
- Spot-Instances werden nicht unterstützt, da die automatische Reparatur für sie nicht einsetzbar ist.

Wenn Sie eine Amazon EC2 EC2-Instance registrieren, ändert AWS OpsWorks Stacks die Sicherheitsgruppen oder Regeln der Instance nicht. Stellen Sie sicher, dass die Sicherheitsgruppenregeln der Instance den folgenden AWS OpsWorks Stacks-Anforderungen entsprechen.

Regeln für eingehenden Datenverkehr

Regeln für eingehenden Datenverkehr sollten Folgendes zulassen:

- SSH-Anmeldung
- Datenverkehr von den entsprechenden Layern.

  Beispielsweise lässt ein Datenbankserver in der Regel eingehenden Datenverkehr von den Anwendungsserver-Layern des Stacks zu.
- Datenverkehr zu den entsprechenden Ports.

Beispielsweise lassen Instances eines Anwendungsservers in der Regel den gesamten eingehenden Datenverkehr zu den Ports 80 (HTTP) und 443 (HTTPS) zu.

Regeln für ausgehenden Datenverkehr

Regeln für ausgehenden Datenverkehr sollten Folgendes zulassen:

- Datenverkehr von Anwendungen, die auf der Instance ausgeführt werden, an den AWS OpsWorks Stacks-Dienst.
- Datenverkehr für den Zugriff auf AWS-Ressourcen wie Amazon S3 von Anwendungen aus, die die AWS-API verwenden.

Eine gängige Methode ist, keine Regeln für den ausgehenden Datenverkehr – und somit auch keinerlei Einschränkungen für diesen – festzulegen.

Installieren und Konfigurieren der AWS CLI

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Bevor Sie Ihre erste Instanz registrieren, müssen Sie Version 1.16.180 von AWS CLI oder neuer auf dem Computer ausführen, von dem aus Sie die Instance ausführen. `register` Die Installationsdetails hängen vom Betriebssystem Ihrer Workstation ab. Weitere Informationen zur Installation von finden Sie unter Installation der AWS-Befehlszeilenschnittstelle und Konfiguration der AWS-Befehlszeilenschnittstelle. AWS CLI Um zu überprüfen, welche Version der AWS CLI Sie ausführen, geben Sie in einer Shell-Sitzung `aws --version` ein.

> **Note**
>
> AWS Tools for PowerShell enthält zwar das `Register-OpsInstance`Cmdlet, das die `register` API-Aktion aufruft, wir empfehlen jedoch, den `register` Befehl stattdessen AWS CLI mit dem auszuführen.

Sie müssen „`register`“ mit den entsprechenden Berechtigungen ausführen. Sie können Berechtigungen mithilfe einer IAM-Rolle oder — weniger optimal — durch die Installation von Benutzeranmeldedaten mit den entsprechenden Berechtigungen auf der zu registrierenden Workstation oder Instance erhalten. Anschließend können Sie die `register` mit diesen Anmeldeinformationen ausführen, wie weiter unten beschrieben. Geben Sie Berechtigungen an, indem Sie dem Benutzer oder der Rolle eine IAM-Richtlinie zuordnen. Denn `register` Sie verwenden entweder die `AWSOpsWorksRegisterCLI_OnPremises` Richtlinien `AWSOpsWorksRegisterCLI_EC2` oder, die Berechtigungen zur Registrierung von Amazon EC2- bzw. lokalen Instances gewähren.

> **Note**
>
> Wenn Sie `register` auf einer Amazon EC2 EC2-Instance arbeiten, sollten Sie idealerweise eine IAM-Rolle verwenden, um Anmeldeinformationen bereitzustellen. Weitere Informationen zum Anhängen einer IAM-Rolle an eine bestehende Instance finden Sie unter Anhängen einer IAM-Rolle an eine Instance oder Ersetzen einer IAM-Rolle im Amazon EC2 EC2-Benutzerhandbuch.

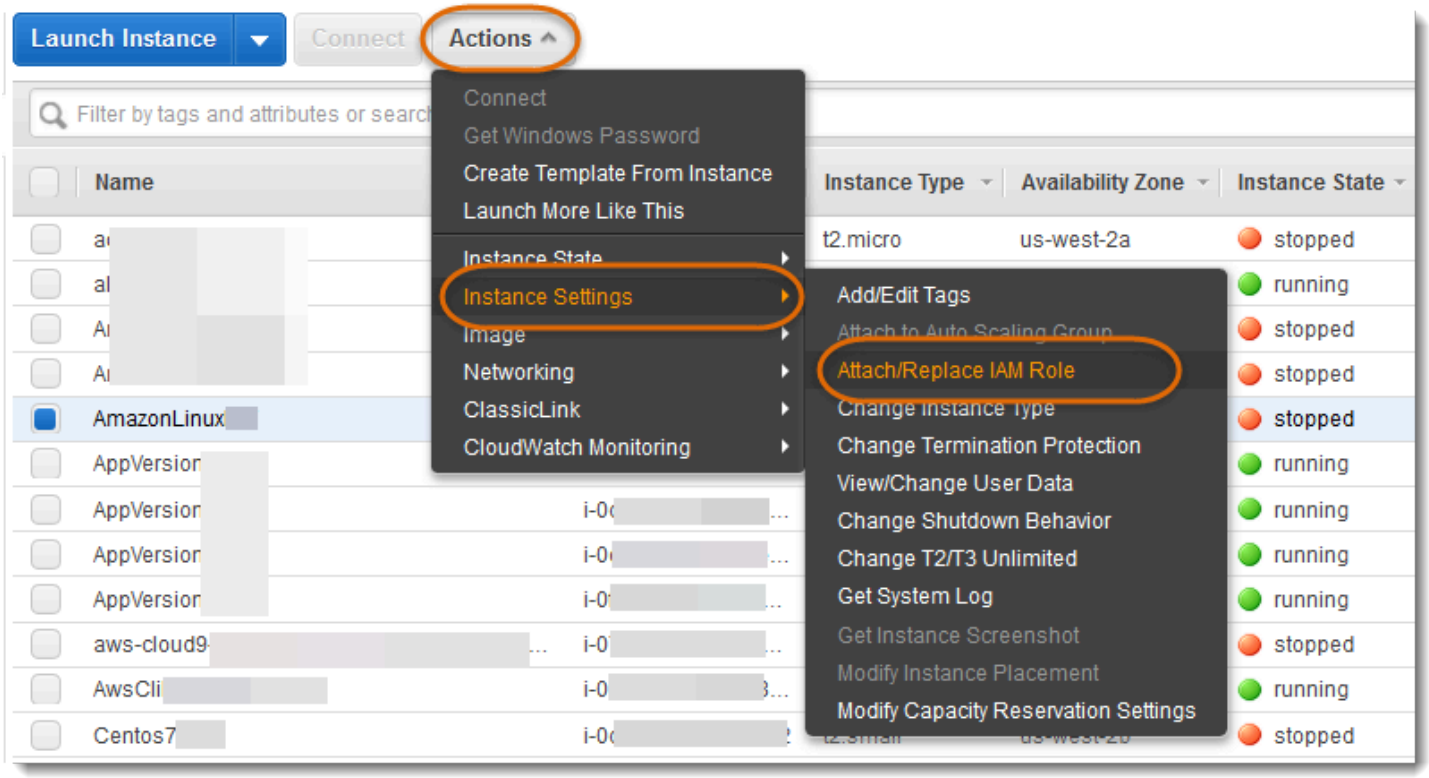
Beispiel-Codeausschnitte der `AWSOpsWorksRegisterCLI_EC2`- und `AWSOpsWorksRegisterCLI_OnPremises`-Richtlinien finden Sie unter Instance-Registrierungsrichtlinien. Weitere Informationen zum Erstellen und Verwalten von AWS-Anmeldeinformationen finden Sie unter AWS-Sicherheitsanmeldeinformationen.

Themen

- Verwenden einer IAM-Rolle
- Verwenden von installierten Anmeldeinformationen

Verwenden einer IAM-Rolle

Wenn Sie den Befehl von der Amazon EC2 EC2-Instance aus ausführen, die Sie registrieren möchten, besteht die bevorzugte Strategie für die Bereitstellung von Anmeldeinformationen `register` darin, eine IAM-Rolle zu verwenden, der die `AWSOpsWorksRegisterCLI_EC2` Richtlinie oder gleichwertige Berechtigungen zugeordnet sind. Bei dieser Methode ist es nicht erforderlich, die Anmeldeinformationen auf der Instance zu installieren. Eine Möglichkeit besteht darin, wie in der folgenden Abbildung dargestellt, den Befehl Attach/Replace IAM Role (IAM-Rolle anfügen/ersetzen) in der EC2-Konsole zu verwenden.



Weitere Informationen zum Anhängen einer IAM-Rolle an eine bestehende Instance finden Sie unter Anhängen einer IAM-Rolle an eine Instance oder Ersetzen einer IAM-Rolle im Amazon EC2 EC2-Benutzerhandbuch. Für Instances, die mit einem Instance-Profil gestartet wurden (empfohlen), fügen Sie den `--use-instance-profile`-Switch zu Ihrem `register`-Befehl hinzu, um Anmeldeinformationen anzugeben; verwenden Sie nicht den `--profile`-Parameter.

Wenn die Instance ausgeführt wird und ihr eine Rolle zugeordnet ist, können Sie die erforderlichen Berechtigungen erteilen, indem Sie der Rolle die `AWSOpsWorksRegisterCLI_EC2`-Richtlinie zuweisen. Die Rolle stellt die Standardanmeldeinformationen für die Instance bereit. Sofern Sie keine Anmeldeinformationen auf der Instance installiert haben, werden die Rolle und deren Berechtigungen von `register` automatisch übernommen.

> **Important**
>
> Es wird empfohlen, keine Anmeldeinformationen auf der Instance zu installieren. Die Rolle der Instance stellt nicht nur ein Sicherheitsrisiko dar, sondern befindet sich auch am Ende der Standardanbieterkette, AWS CLI anhand derer sie die Standardanmeldedaten ausfindig macht. Somit könnten installierte Anmeldeinformationen Vorrang vor der Rolle haben, sodass

`register` ggf. nicht über die erforderlichen Berechtigungen verfügt. Weitere Informationen finden Sie unter Erste Schritte mit AWS CLI.

Wird eine Instance ohne Rolle ausgeführt, müssen Sie die Anmeldeinformationen mit den erforderlichen Berechtigungen auf der Instance selbst installieren, wie beschrieben unter Verwenden von installierten Anmeldeinformationen. Es ist empfohlen, einfacher und weniger fehleranfällig, Instances zu verwenden mit einem Instance-Profil gestartet werden.

Verwenden von installierten Anmeldeinformationen

Es gibt mehrere Möglichkeiten, Benutzeranmeldeinformationen auf einem System zu installieren und sie einem AWS CLI Befehl zur Verfügung zu stellen. Im Folgenden wird ein Ansatz beschrieben, der nicht mehr empfohlen wird, jedoch verwendet werden kann, wenn Sie EC2-Instances registrieren, die ohne ein Instance-Profil gestartet wurden. Sie können auch die Anmeldeinformationen eines vorhandenen -Benutzers nutzen, sofern die zugeordneten Richtlinien die erforderlichen Berechtigungen erteilen. Weitere Informationen, darunter andere Möglichkeiten zur Installation von Anmeldeinformationen, finden Sie unter Konfigurations- und Anmeldeinformationsdateien.

So verwenden Sie installierte Anmeldeinformationen

1. Erstellen Sie einen IAM-Benutzer und speichern Sie die Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel an einem sicheren Ort.

   > **Warning**
   >
   > IAM-Benutzer verfügen über langfristige Anmeldeinformationen, was ein Sicherheitsrisiko darstellt. Um dieses Risiko zu minimieren, empfehlen wir, diesen Benutzern nur die Berechtigungen zu gewähren, die sie für die Ausführung der Aufgabe benötigen, und diese Benutzer zu entfernen, wenn sie nicht mehr benötigt werden.

2. Hängen Sie die AWSOpsWorksRegisterCLI_OnPremises Richtlinie an den Benutzer an. Ggf. können Sie auch eine Richtlinie mit weiteren Berechtigungen zuweisen, allerdings müssen die Berechtigungen von `AWSOpsWorksRegisterCLI_OnPremises` enthalten sein.
3. Erstellen Sie in der `credentials`-Datei des Systems ein Profil für den Benutzer. Die Datei finden Sie unter `~/.aws/credentials` (Linux, Unix und OS X) oder `C:\Users\`*`User_Name`*`\.aws\credentials` (Windows-Systeme). Die Datei enthält ein oder mehrere

Profile im folgenden Format, von denen jedes die Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel eines Benutzers enthält.

```
[profile_name]
aws_access_key_id = access_key_id
aws_secret_access_key = secret_access_key
```

Ersetzen Sie die Werte access_key_id und secret_access_key durch die IAM-Anmeldeinformationen, die Sie zuvor gespeichert haben. Sie können einen beliebigen Namen als Profilnamen angeben (mit zwei Einschränkungen: der Name muss eindeutig sein und das Standardprofil muss `default` heißen). Sie können auch ein vorhandenes Profil verwenden, sofern es über die notwendigen Berechtigungen verfügt.

4. Geben Sie den Profilnamen im `--profile`-Parameter des `register`-Befehls an. Der Befehl `register` wird mit den Berechtigungen ausgeführt, die den zugeordneten Anmeldeinformationen erteilt wurden.

   Sie können `--profile` auch auslassen. In dem Fall wird `register` mit den Standardanmeldeinformationen ausgeführt. Beachten Sie, dass es sich dabei nicht zwangsläufig um die Anmeldeinformationen des Standardprofils handelt. Sie müssen daher sicherstellen, dass die Standardanmeldeinformationen über die erforderlichen Berechtigungen verfügen. Weitere Informationen darüber, wie der die Standardanmeldedaten AWS CLI bestimmt, finden Sie unter Konfiguration der AWS-Befehlszeilenschnittstelle.

Registrieren der Instance

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Funktion wird nur für Linux-Stacks unterstützt.

Sie registrieren eine Instance, indem Sie den Befehl AWS CLI `register` von der Workstation oder auf der Instance ausführen. Die einfachste Möglichkeit dafür bietet der Registrierungsassistent der AWS OpsWorks Stacks-Konsole, der die Erstellung der Befehlszeichenfolge vereinfacht. Sofern Sie mit dem Registrierungsverfahren vertraut sind, können Sie ggf. den Assistenten überspringen und den Befehl `register` ausführen.

Nachfolgend wird beschrieben, wie Sie mit dem Registrierungsassistenten eine Instance für einen vorhandenen Stack registrieren.

> **Note**
>
> Um eine Instance bei einem neuen Stack zu registrieren, kannst du dies tun, indem du im AWS OpsWorks Stacks-Dashboard die Option Instanzen registrieren auswählst. Der gestartete Assistent ist mit dem für vorhandene Stacks identisch und weist eine weitere Seite für die Konfiguration des neuen Stacks auf.

So verwenden Sie den Registrierungsassistenten zum Registrieren einer Instance

1. Erstellen Sie in der AWS OpsWorks -Konsole einen Stack oder öffnen Sie einen vorhandenen Stack.
2. Wählen Sie im Navigationsbereich Instances und anschließend register an instance aus.
3. Geben Sie auf der Seite Choose an Instance Type an, ob Sie eine Amazon EC2- oder eine lokale Instance registrieren möchten:
   - Wenn Sie eine Amazon EC2 EC2-Instance registrieren, wählen Sie Weiter: Select Instances.
   - Wenn Sie eine lokale Instance registrieren, wählen Sie Weiter: AWS-CLI installieren und fahren Sie dann mit Schritt 5 fort.
4. Wenn Sie eine Amazon EC2 EC2-Instance registrieren, öffnen Sie die Seite Select Instances, um die zu registrierende Instance auszuwählen. AWS OpsWorks Stacks sammelt die Informationen, die zur Erstellung des Befehls benötigt werden. Wenn Sie fertig sind, klicken Sie auf Next:Install AWS CLI.

5. Die Instanz, auf der Sie ausführen möchten, `register` muss Version 1.16.180 von oder neuer ausführen. AWS CLI Wenn Sie die AWS CLI installieren oder aktualisieren müssen, finden Sie auf der Seite des Registrierungsassistenten Links zu Installations- und Konfigurationsanweisungen. Überprüfen Sie die AWS CLI -Installation und geben Sie anschließend an, ob der Befehl auf der zu registrierenden Instance oder von einer separaten Workstation ausgeführt werden soll. Wählen Sie dann Next: Register Instances (Weiter: Instances registrieren) aus.
6. Auf der Seite Register Instances wird eine Vorlage für eine `register`-Befehlszeichenfolge mit den von Ihnen ausgewählten Optionen angezeigt. Wenn Sie beispielsweise eine Amazon EC2 EC2-Instance von einer separaten Workstation aus registrieren, ähnelt die Standardvorlage der folgenden.

```
aws opsworks register --infrastructure-class ec2 --region us-west-2
  --stack-id 247be7ea-3551-4177-9524-1ff804f453e3 --ssh-username [username] i-
f1245d10
```

> **Important**
>
> Der IAM-Benutzer, der während des Registrierungsprozesses erstellt wird, ist während der gesamten Lebensdauer einer registrierten Instance erforderlich. Das Löschen des Benutzers führt dazu, dass der AWS OpsWorks Stacks-Agent nicht mit dem Dienst kommunizieren kann. Um Probleme bei der Verwaltung registrierter Instanzen zu vermeiden, falls der Benutzer versehentlich gelöscht wird, fügen Sie Ihrem `register` Befehl den `--use-instance-profile` Parameter hinzu, um stattdessen das integrierte Instanzprofil der Instanz zu verwenden. Durch das Hinzufügen des `--use-instance-profile` Parameters wird außerdem verhindert, dass Fehler auftreten, wenn Sie die AWS Kontozugriffsschlüssel alle 90 Tage wechseln (eine empfohlene bewährte Methode), da auf diese Weise Diskrepanzen zwischen den für den AWS OpsWorks Agenten verfügbaren Zugriffsschlüsseln und den erforderlichen IAM-Benutzern vermieden werden.

Wenn Sie Ich verwende SSH-Schlüssel auf Ja setzen, fügt AWS OpsWorks Stacks das `--ssh-private-key` Argument zur Zeichenfolge hinzu, mit der Sie eine private SSH-Schlüsseldatei angeben können.

> **Note**
>
> Wenn Sie sich mit einem Passwort anmelden **`register`** möchten, setzen Sie Ich verwende SSH-Schlüssel auf Nein. Beim Ausführen `register` werden Sie zur Eingabe des Kennworts aufgefordert.

Kopieren Sie diese Zeichenfolge in einen Texteditor und nehmen Sie die erforderlichen Änderungen vor. Beachten Sie Folgendes:

- Der Text in Klammern gibt Informationen vor, die Sie bereitstellen müssen (z. B. den Speicherort der SSH-Schlüsseldatei).
- Die Vorlage geht davon aus, dass `register` mit den AWS-Standardanmeldeinformationen ausgeführt wird. Ist das nicht Fall, fügen Sie der Befehlszeichenfolge ein `--profile`-Argument hinzu und geben Sie den Profilnamen an, der für die Anmeldeinformationen verwendet werden soll.

Für andere Szenarien sind möglicherweise weitere Änderungen am Befehl erforderlich. Eine Beschreibung der verfügbaren `register`-Argumente sowie andere Methoden zur Erstellung der Befehlszeichenfolge finden Sie unter Verwenden des Befehls `register`. Um die Dokumentation zum Befehl anzuzeigen, führen Sie `aws opsworks help register` in der Befehlszeile aus. Einige Beispiele für die Befehlszeichenfolge finden Sie unter "register"-Befehlsbeispiele.

7. Nachdem Sie die Bearbeitung der Befehlszeichenfolge beendet haben, öffnen Sie ein Terminalfenster auf Ihrer Workstation oder melden Sie sich per SSH an der Instance an. Führen Sie dann den Befehl aus. Der gesamte Vorgang dauert in der Regel etwa fünf Minuten, in denen die Instance den Status Registering aufweist.
8. Nach Beendigung des Vorgangs klicken Sie auf Done. Die Instance hat jetzt den Status Registered und wird auf der Seite Instances des Stacks als nicht zugeordnete Instance aufgeführt.

Der Befehl `register` hat folgende Auswirkungen:

1. Bei Ausführung von "`register`" auf einer Workstation erfolgt als Erstes die Anmeldung des Befehls (über SSH) an der zu registrierenden Instance.

Die restlichen Schritte erfolgen auf der Instance und sind unabhängig davon, wo der Befehl ausgeführt wird, identisch.

2. Lädt das AWS OpsWorks Stacks-Agentenpaket von Amazon S3 herunter.
3. Der Agent und dessen Abhängigkeiten wie z. B. AWS SDK für Ruby werden entpackt und installiert.
4. Folgendes wird erstellt:
   - Ein IAM-Benutzer, der den Agenten mit dem AWS OpsWorks Stacks-Service bootet, um eine sichere Kommunikation zu gewährleisten.

     Die Benutzerberechtigungen lassen nur die `opsworks:RegisterInstance`-Aktion zu und sind nach 15 Minuten abgelaufen.
   - Eine IAM-Gruppe für den Stack, die die Benutzer der registrierten Instances enthält.
5. Erzeugt ein RSA-Schlüsselpaar und sendet den öffentlichen Schlüssel an AWS OpsWorks Stacks.

   Dieses Schlüsselpaar wird zur Verschlüsselung der Kommunikation zwischen Agent und AWS OpsWorks Stacks eingesetzt.
6. Registriert die Instanz bei AWS OpsWorks Stacks. Anschließend führt der Stack einige Rezepte für die Ersteinrichtung aus, um die Instance zu konfigurieren, darunter z. B.:
   - Die Hosts-Datei der Instance wird überschrieben.

     Durch die Registrierung der Instanz haben Sie die Benutzerverwaltung an AWS OpsWorks Stacks übergeben, das über eine eigene Hosts-Datei verfügen muss, um die SSH-Anmeldeberechtigungen zu kontrollieren.
   - Bei Amazon EC2 EC2-Instances umfasst die Ersteinrichtung auch die Registrierung aller angehängten Amazon EBS-Volumes oder Elastic IP-Adressen beim Stack.

     Sie müssen sicherstellen, dass die Amazon EBS-Volumes nicht an reservierten Mount-Points gemountet werden, einschließlich `/var/www` aller Mount-Points, die von den Layern der Instance reserviert sind. Weitere Informationen zum Verwalten von Stack-Ressourcen finden Sie unter Ressourcenmanagement. Weitere Informationen zu Mounting-Punkten für Layer finden Sie unter AWS OpsWorks Stacks-Ebenenreferenz.

Eine vollständige Beschreibung der Konfigurationsänderungen im Rahmen der Ersteinrichtung finden Sie unter Konfigurationsänderungen im Rahmen der Ersteinrichtung.

> **Note**
>
> Das Betriebssystem einer registrierten Instance wird im Rahmen der Ersteinrichtung nicht aktualisiert. Dieser Schritt muss von Ihnen ausgeführt werden. Weitere Informationen finden Sie unter Verwalten von Sicherheitsupdates.

Verwenden des Befehls **`register`**

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Funktion wird nur für Linux-Stacks unterstützt.

Um eine Instance registrieren zu können, müssen Sie mindestens Version 1.16.180 der AWS CLI ausführen. Nachfolgend finden Sie die allgemeine Syntax für den Befehl "`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]
```

Die folgenden Argumente sind allen AWS CLI Befehlen gemeinsam.

`--profile`

(Optional) Der Profilname mit den Anmeldeinformationen. Falls Sie dieses Argument nicht angeben, wird der Befehl mit den Standardanmeldeinformationen ausgeführt. Weitere Informationen darüber, wie der die Standardanmeldedaten AWS CLI bestimmt, finden Sie unter Konfiguration der AWS-Befehlszeilenschnittstelle.

`--region`

(Optional) Die Region des AWS OpsWorks Stacks-Serviceendpunkts. Stellen Sie nicht `--region` die Region des Stacks ein. AWS OpsWorks Stacks bestimmt die Region des Stacks automatisch anhand der Stack-ID.

> **Note**
>
> Wenn Ihre Standardregion bereits festgelegt ist, können Sie dieses Argument weglassen. Weitere Informationen zur Angabe einer Standardregion finden Sie unter Konfiguration der AWS-Befehlszeilenschnittstelle.

Verwenden Sie die folgenden Argumente sowohl für Amazon EC2- als auch für lokale Instances.

`--infrastructure-class`

(Erforderlich) Dieser Parameter muss entweder auf `ec2` oder gesetzt werden`on-premises`, um anzugeben, ob Sie eine Amazon EC2- oder eine lokale Instance registrieren.

`--stack-id`

(Erforderlich) Die ID des Stacks, für den die Instance registriert werden soll.

> **Note**
>
> Klicken Sie auf der Seite Stack auf Settings, um die Stack-ID zu bestimmen. Die Stack-ID trägt die Bezeichnung OpsWorks ID und ist eine GUID, die ungefähr so aussieht. `ad21bce6-7623-47f1-bf9d-af2affad8907`

Argumente für die SSH-Anmeldung

Mit den folgenden Argumenten geben Sie an, wie die Anmeldung von `register` an der Instance erfolgt.

`--local`

(Optional) Mit diesem Argument registrieren Sie die Instance, auf der Sie den Befehl ausführen.

In diesem Fall muss keine Anmeldung von `register` an der Instance erfolgen.

`--ssh-private-key` und **`--ssh-username`**

(Optional) Verwenden Sie diese Argumente, wenn Sie eine Instance von einer separaten Workstation registrieren und den Benutzernamen oder die private Schlüsseldatei explizit angeben möchten.

- `--ssh-username`— Verwenden Sie dieses Argument, um einen SSH-Benutzernamen anzugeben.

  Falls Sie `--ssh-username` nicht angeben, verwendet `ssh` den Standardbenutzernamen.
- `--ssh-private-key`— Verwenden Sie dieses Argument, um explizit eine private Schlüsseldatei anzugeben.

  Falls Sie `--ssh-private-key` nicht angeben, versucht `ssh`, sich mit Authentifizierungsmethoden anzumelden, die kein Passwort erfordern, einschließlich der Verwendung des privaten Schlüssels. Wird keine dieser Methoden unterstützt, fragt `ssh` nach dem Passwort. Weitere Informationen darüber, wie die Authentifizierung mit `ssh` erfolgt, finden Sie unter The Secure Shell (SSH) Authentication Protocol.

`--override-ssh`

(Optional) Verwenden Sie dieses Argument, wenn Sie die Instance von einer separaten Workstation registrieren und eine benutzerdefinierte `ssh`-Befehlszeichenfolge angeben möchten. Vom Befehl `register` wird diese Befehlszeichenfolge für die Anmeldung an der registrierten Instance verwendet.

Weitere Informationen zu `ssh` finden Sie unter SSH.

`--override-hostname`

(Optional) Gibt einen Hostnamen für die Instanz an, der nur von AWS OpsWorks Stacks verwendet wird. Der Standardwert ist der Name des Instance-Hosts.

`--debug`

(Optional) Bietet im Falle einer fehlgeschlagenen Registrierung Debugging-Informationen. Informationen zur Problembehebung finden Sie unter Fehlerbehebung bei der Instance-Registrierung.

`--use-instance-profile`

(Optional, aber für Amazon EC2 EC2-Instances dringend empfohlen) Ermöglicht es dem `register` Befehl, ein angehängtes Instance-Profil zu verwenden, anstatt einen IAM-Benutzer zu erstellen. Das Hinzufügen dieses Parameters kann dazu beitragen, Fehler zu vermeiden, die auftreten, wenn Sie versuchen, eine registrierte Instance zu verwalten, obwohl der IAM-Benutzer versehentlich gelöscht wurde.

> **Important**
>
> Der IAM-Benutzer, der während des Registrierungsprozesses erstellt wird, ist während der gesamten Lebensdauer einer registrierten Instance erforderlich. Das Löschen des Benutzers führt dazu, dass der AWS OpsWorks Stacks-Agent nicht mit dem Dienst kommunizieren kann. Um Probleme bei der Verwaltung registrierter Instanzen zu vermeiden, falls der Benutzer versehentlich gelöscht wird, fügen Sie Ihrem `register` Befehl den `--use-instance-profile` Parameter hinzu, um stattdessen das integrierte Instanzprofil der Instanz zu verwenden. Durch das Hinzufügen des `--use-instance-profile` Parameters wird außerdem verhindert, dass Fehler auftreten, wenn Sie die AWS Kontozugriffsschlüssel alle 90 Tage wechseln (eine empfohlene bewährte Methode), da auf diese Weise Diskrepanzen zwischen den für den AWS OpsWorks Agenten verfügbaren Zugriffsschlüsseln und den erforderlichen Benutzern vermieden werden.

Ziel

(Bedingt) Wenn Sie diesen Befehl von einer Workstation ausführen, gibt der letzte Wert in der Befehlszeichenfolge das Registrierungsziel auf eine der folgenden Weisen an.

- Die öffentliche IP-Adresse der Instance.
- Der Name des Instance-Hosts.
- Für Amazon EC2 EC2-Instances die Instance-ID.

  AWS OpsWorks Stacks verwendet die Instance-ID, um die Instance-Konfiguration abzurufen, einschließlich der öffentlichen IP-Adresse der Instance. Standardmäßig verwendet AWS

OpsWorks Stacks diese Adresse, um die `ssh` Befehlszeichenfolge zu erstellen, mit der es sich bei der Instanz anmeldet. Falls die Verbindung zu einer privaten IP-Adresse hergestellt werden soll, muss mit `--override-ssh` eine benutzerdefinierte Befehlszeichenfolge bereitgestellt werden. Ein Beispiel finden Sie unter Registrieren einer lokalen Instance von einer Workstation.

> **Note**
>
> Bei Angabe eines Host-Namens ist `ssh` davon abhängig, dass der DNS-Server den Namen zu einer bestimmten Instance auflöst. Falls Sie nicht sicher sind, dass der Host-Name eindeutig ist, können Sie anhand von `ssh` überprüfen, ob der Host-Name zur richtigen Instance aufgelöst wird.

Wenn Sie diesen Befehl auf der zu registrierenden Instance ausführen, lassen Sie die Instance-ID weg und verwenden stattdessen das `--local`-Argument.

Die folgenden Argumente gelten nur für lokale Instances.

`--override-public-ip`

(Optional) AWS OpsWorks Stacks zeigt die angegebene Adresse als öffentliche IP-Adresse der Instanz an. Die öffentliche IP-Adresse der Instance wird nicht geändert. Wenn ein Benutzer jedoch die Konsole verwendet, um eine Verbindung zur Instance herzustellen, z. B. indem er die Adresse auf der Seite Instances auswählt, verwendet AWS OpsWorks Stacks die angegebene Adresse. AWS OpsWorks Stacks bestimmt automatisch den Standardwert des Arguments.

`--override-private-ip`

(Optional) AWS OpsWorks Stacks zeigt die angegebene Adresse als private IP-Adresse der Instanz an. Die private IP-Adresse der Instanz wird dadurch nicht geändert. AWS OpsWorks Stacks bestimmt automatisch den Standardwert des Arguments.

"register"-Befehlsbeispiele

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir

empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Funktion wird nur für Linux-Stacks unterstützt.

In diesem Abschnitt werden einige Beispiele für `register`-Befehlszeichenfolgen beschrieben.

Registrieren Sie eine Amazon EC2 EC2-Instance von einer Workstation aus

Das folgende Beispiel registriert eine Amazon EC2 EC2-Instance von einer Workstation aus. Die Befehlszeichenfolge verwendet Standardanmeldedaten und identifiziert die Instance anhand ihrer Amazon EC2 EC2-Instance-ID. Sie können dieses Beispiel auch für lokale Instances verwenden, indem Sie `ec2` in `on-premises` ändern.

```
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
```

Registrieren einer lokalen Instance von einer Workstation

Im folgenden Beispiel wird eine lokale Instance von einer separaten Workstation registriert. Die Befehlszeichenfolge verwendet die Standardanmeldeinformationen und meldet sich mit der angegebenen `ssh`-Befehlszeichenfolge an der Instance an. Falls für die Instance ein Passwort erforderlich ist, werden Sie von `register` zur Eingabe aufgefordert. Sie können das Beispiel für Amazon EC2 EC2-Instances verwenden, indem Sie `on-premises` zu `ec2` wechseln.

```
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**
>
> Sie können `--override-ssh` damit eine beliebige benutzerdefinierte SSH-Befehlszeichenfolge angeben. AWS OpsWorks Stacks verwendet dann die angegebene Zeichenfolge, um sich bei der Instanz anzumelden, anstatt eine Befehlszeichenfolge zu erstellen. Ein weiteres Beispiel finden Sie unter Registrieren einer Instance mithilfe einer benutzerdefinierten SSH-Befehlszeichenfolge.

Registrieren einer Instance mithilfe einer benutzerdefinierten SSH-Befehlszeichenfolge

Das folgende Beispiel registriert eine lokale Instanz von einer Workstation aus und verwendet das `--override-ssh` Argument, um einen benutzerdefinierten SSH-Befehl anzugeben, der für die Anmeldung bei der Instanz `register` verwendet wird. In diesem Beispiel erfolgt die Anmeldung von `sshpass` mit einem Benutzernamen und einem Passwort, aber Sie können auch eine gültige `ssh`-Befehlszeichenfolge spezifizieren.

```
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"
```

Registrieren einer Instance mittels der **`register`**-Ausführung auf der Instance

Das folgende Beispiel zeigt, wie Sie eine Amazon EC2 EC2-Instance registrieren, indem Sie sie von der Instance selbst `register` aus ausführen. Die Befehlszeichenfolge hängt in Bezug auf die Berechtigungen von den Standardanmeldeinformationen ab. Um das Beispiel für eine lokale Instance zu verwenden, wechseln Sie `--infrastructure-class` zu. `on-premises`

```
aws opsworks register \
  --region us-west-2 \
  --infrastructure-class ec2 \
  --stack-id ad21bce6-7623-47f1-bf9d-af2affad8907 \
  --local
```

Registrieren einer Instance mit einer privaten IP-Adresse

Standardmäßig wird von `register` die öffentliche IP-Adresse der Instance für die Anmeldung an der Instance verwendet. Wenn Sie eine Instance mit einer privaten IP-Adresse registrieren möchten (z. B. eine Instance im privaten Subnetz einer VPC), müssen Sie `--override-ssh` verwenden, um eine benutzerdefinierte `ssh`-Befehlszeichenfolge anzugeben.

```
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
```

Instance-Registrierungsrichtlinien

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Die Richtlinien `AWSOpsWorksRegisterCLI_EC2` und `AWSOpsWorksRegisterCLI_OnPremises` bieten die richtigen Berechtigungen zum Registrieren von EC2-Instances bzw. lokalen Instances. Sie fügen Ihren IAM-Benutzer hinzu`AWSOpsWorksRegisterCLI_EC2`, um EC2-Instances zu registrieren, aber fügen Sie Ihren Benutzer hinzu, `AWSOpsWorksRegisterCLI_OnPremises` um lokale Instances zu registrieren. Um diese Richtlinien verwenden zu können, müssen Sie mindestens Version 1.16.180 von oder neuer ausführen. AWS CLI

Die Richtlinie **`AWSOpsWorksRegisterCLI_EC2`**

Fügen Sie Ihrem Benutzer hinzu`AWSOpsWorksRegisterCLI_EC2`, um EC2-Instances zu registrieren. Sie sollten dieses Profil verwenden, wenn Sie vorhaben, ausschließlich EC2-Instances zu registrieren. Wenn Sie dieses Profil verwenden, werden Berechtigungen durch das Instance-Pofil der EC2-Instance erteilt.

```
{
      "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": [
            "*"
          ]
        }
      ]
    }
```

Die Richtlinie **AWSOpsWorksRegisterCLI_OnPremises**

Fügen Sie `AWSOpsWorksRegisterCLI_OnPremises` Ihren Benutzer hinzu, um lokale Instances zu registrieren. Diese Richtlinie umfasst z. B. IAM-Berechtigungen, aber die Ressourcen`AttachUserPolicy`, für die diese Berechtigungen gelten, sind eingeschränkt.

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

(Veraltet) Die Richtlinie **`AWSOpsWorksRegisterCLI`**

> **Important**
>
> Die Richtlinie `AWSOpsWorksRegisterCLI` ist veraltet und kann nicht zum Registrieren von neuen Instances verwendet werden. Sie ist nur zwecks Abwärtskompatibilität auf Instances verfügbar, die bereits registriert wurden. Die `AWSOpsWorksRegisterCLI` Richtlinie umfasst viele IAM-Berechtigungen`CreateUser`, darunter`PutUserPolicy`, und. `AddUserToGroup` Weil es sich hierbei um die Admin-Berechtigungen handelt, sollten Sie die Richtlinie `AWSOpsWorksRegisterCLI` nur vertrauenswürdigen Benutzern zuweisen.

Verwalten von registrierten Instances

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Funktion wird nur für Linux-Stacks unterstützt.

Wenn Sie eine Instanz registrieren, wird sie zu einer AWS OpsWorks Stacks-Instanz, und Sie können sie auf die gleiche Weise verwalten wie mit Stacks erstellte Instanzen. AWS OpsWorks Es gibt zwei wesentliche Unterschiede:

- Registrierte Instances müssen keinem Layer zugewiesen werden.
- Sie können die Registrierung einer Instance aufheben und diese wieder direkt kontrollieren.

Nachdem Sie eine Instance registriert haben, befindet sie sich im Status Registriert. AWS OpsWorks Stacks bietet die folgenden Verwaltungsfunktionen für alle registrierten Instances:

- Integritätsprüfungen — AWS OpsWorks Stacks überwacht den Agenten, um zu bewerten, ob die Instanz weiterhin funktioniert.

  Wenn eine Instance eine Zustandsprüfung nicht besteht, heilt AWS OpsWorks Stacks registrierte Amazon EC2 EC2-Instances automatisch und ändert den Status registrierter On-Premises-Instances in. `connection lost`
- CloudWatch Überwachung — Die Überwachung ist für CloudWatch registrierte Instances aktiviert.

  Sie können Metriken wie CPU-Auslastung und verfügbaren Arbeitsspeicher überwachen und optional eine Benachrichtigung erhalten, wenn eine Metrik einen festgelegten Schwellenwert überschreitet.
- Benutzerverwaltung — AWS OpsWorks Stacks bietet eine einfache Möglichkeit, festzulegen, welche Benutzer auf die Instanz zugreifen können und welche Operationen sie ausführen dürfen. Weitere Informationen finden Sie unter Verwalten von Benutzerberechtigungen.
- Ausführung von Rezepten — Sie können den Stack-Befehl Execute Recipes verwenden, um Chef-Rezepte auf der Instanz auszuführen.
- Betriebssystem-Updates — Sie können den Stack-Befehl Update Dependencies verwenden, um das Betriebssystem der Instanz zu aktualisieren.

Um alle Vorteile der AWS OpsWorks Stacks-Verwaltungsfunktionen nutzen zu können, können Sie die Instanz einer Ebene zuweisen. Weitere Informationen finden Sie unter Zuweisen einer registrierten Instance zu einem Layer.

Es gibt Unterschiede zwischen der Art und Weise, wie AWS OpsWorks Stacks Amazon EC2 verwaltet, und lokalen Instances.

Amazon EC2 EC2-Instances

- Wenn Sie eine registrierte Amazon EC2 EC2-Instance beenden, beendet AWS OpsWorks Stacks Instances, die vom Instance-Speicher unterstützt werden, und stoppt Amazon EBS-gestützte Instances.

  Die Instance bleibt für den Stack registriert und den Layern zugewiesen, Sie können sie bei Bedarf also neu starten. Um eine registrierte Instance aus einem Stack zu entfernen, müssen Sie entweder die Registrierung explizit aufheben oder die Instance löschen (damit wird die Registrierung automatisch aufgehoben).
- Wenn Sie eine registrierte Amazon EC2-Instance neu starten oder die Instance ausfällt und automatisch repariert wird, entspricht das Ergebnis dem Stoppen und Neustarten der Instance mithilfe von Amazon EC2. Beachten Sie die folgenden Unterschiede:
  - Instance Store-Backed Instances — AWS OpsWorks Stacks startet eine neue Instance mit demselben AMI.

    Beachten Sie, dass AWS OpsWorks Stacks keine Kenntnis von Vorgängen hat, die Sie vor der Registrierung an der Instance ausgeführt haben, wie z. B. die Installation von Softwarepaketen. Wenn Sie möchten, dass AWS OpsWorks Stacks beim Start Pakete installiert oder andere Konfigurationsaufgaben ausführt, müssen Sie benutzerdefinierte Chef-Rezepte bereitstellen, die die erforderlichen Aufgaben ausführen, und sie den Setup-Ereignissen der entsprechenden Ebenen zuweisen.
  - Amazon EBS-gestützte Instances — AWS OpsWorks Stacks startet eine neue Instance mit demselben AMI und fügt das Root-Volume erneut an, wodurch die Instance auf ihre vorherige Konfiguration zurückgesetzt wird.
- Wenn Sie eine registrierte Amazon EC2 EC2-Instance abmelden, wird sie wieder zu einer regulären Amazon EC2 EC2-Instance.

Lokale Instances

- AWS OpsWorks Stacks können eine registrierte lokale Instance nicht stoppen oder starten.

Das Aufheben der Zuweisung einer registrierten lokalen Instanz löst ein Shutdown-Ereignis aus. Damit werden jedoch nur die Shutdown-Rezepte des zugewiesenen Layers ausgeführt. Sie führen Aufgaben (wie z. B. Services herunterfahren) aus, stoppen jedoch nicht die Instance.

- AWS OpsWorks Stacks können eine registrierte lokale Instanz nicht automatisch reparieren, wenn sie ausfällt, aber die Instanz wird als Verbindungsverlust markiert.
- Lokale Instances können die Elastic Load Balancing-, Amazon EBS- oder Elastic IP-Adressdienste nicht verwenden.

Zuweisen einer registrierten Instance zu einem Layer

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Diese Funktion wird nur für Linux-Stacks unterstützt.

Nachdem Sie eine Instance registriert haben, können Sie diese einem oder mehreren Layern zuweisen. Der Vorteil, einer Ebene eine Instanz zuzuweisen, anstatt sie unzugewiesen zu lassen, besteht darin, dass Sie den Lebenszyklusereignissen der Ebene benutzerdefinierte Rezepte zuweisen können. AWS OpsWorks Stacks führt sie dann automatisch zum richtigen Zeitpunkt aus, und zwar nach den Rezepten der Ebene für dieses Ereignis.

- Sie können jede registrierte Instance einem benutzerdefinierten Layer zuweisen. Ein benutzerdefinierter Layer verfügt über einen minimalen Rezeptsatz, mit dem keinerlei Pakete installiert werden, daher gibt es keine Konflikte mit der bestehenden Instance-Konfiguration.
- Sie können den integrierten Layern von AWS OpsWorks Stacks lokale Instanzen zuweisen.

Jeder integrierter Layer enthält Rezepte, mit denen automatisch ein oder mehrere Pakete installiert werden. Mit den Setup-Rezepten für Java App Server werden beispielsweise Apache und Tomcat installiert. Die Layer-Rezepte können auch andere Vorgänge ausführen, so z. B. Services neu starten und Anwendungen bereitstellen. Bevor Sie einer integrierten Ebene eine lokale Instanz zuweisen, sollten Sie sicherstellen, dass die Rezepte der Ebene keine Konflikte verursachen, z. B. wenn Sie versuchen, eine andere Anwendungsserverversion zu installieren als die, die sich derzeit auf der Instanz befindet. Weitere Informationen finden Sie unter Ebenen und AWS OpsWorks Stacks-Ebenenreferenz.

So weisen Sie eine registrierte Instance einem Layer zu

1. Fügen Sie die zu verwendenden Layer zum Stack hinzu (sofern noch nicht geschehen).
2. Klicken Sie im Navigationsbereich auf Instances und anschließend auf assign in der Spalte Actions der Instance.
3. Wählen Sie die entsprechenden Layer aus und klicken Sie auf Save.

Wenn Sie einer Ebene eine Instanz zuweisen, geht AWS OpsWorks Stacks wie folgt vor.

- Die Einrichtungsrezepte des Layers werden ausgeführt.
- Fügt alle angehängten Elastic IP-Adressen oder Amazon EBS-Volumes zu den Ressourcen des Stacks hinzu.

  Anschließend können Sie AWS OpsWorks Stacks verwenden, um diese Ressourcen zu verwalten. Weitere Informationen finden Sie unter Ressourcenmanagement.

Nachdem sie abgeschlossen sind, befindet sich die Instanz im Online-Status und ist vollständig in den Stack integriert. AWS OpsWorks Stacks führt dann jedes Mal, wenn ein Lebenszyklusereignis eintritt, die dem Layer zugewiesenen Rezepte aus.

Aufheben der Zuweisung einer registrierten Instance

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

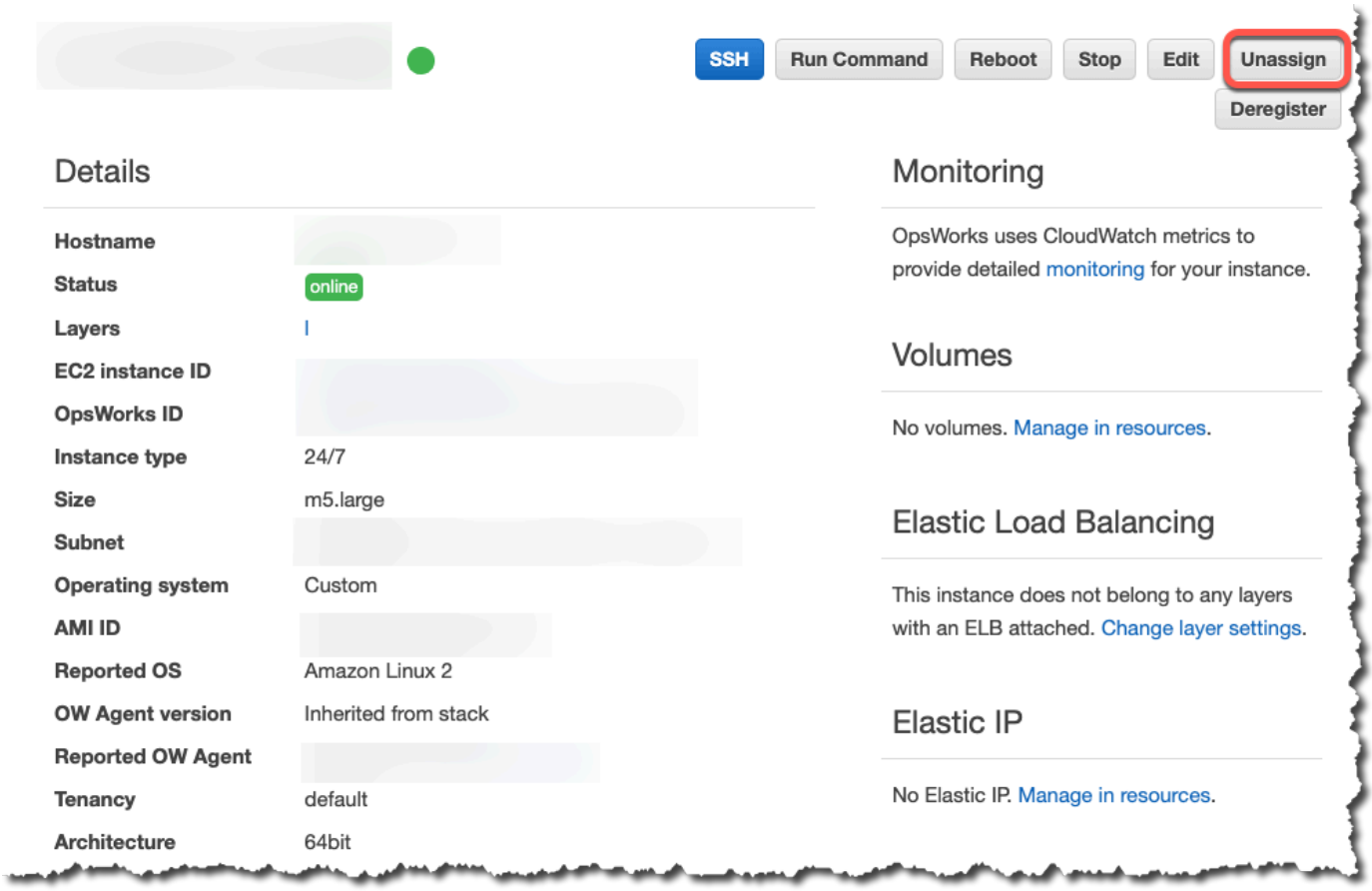
> **Note**
>
> Diese Funktion wird nur für Linux-Stacks unterstützt.

Sie können die Zuweisung einer registrierten Instanz zu ihren Layern aufheben, indem Sie die AWS OpsWorks Konsole oder den AWS CLI SDK-Vorgang verwenden.

Wenn Sie die Zuweisung einer Instanz aufheben, führt AWS OpsWorks Stacks die Shutdown-Rezepte der Ebene auf der Instanz aus. Diese Rezepte führen Aufgaben (wie z. B. Services herunterfahren) aus, stoppen jedoch nicht die Instance. Falls die Instance mehreren Layern zugewiesen ist, wird die Zuweisung für jeden Layer aufgehoben. Sie können die Zuweisung einer Instance nicht nur für einige der zugewiesenen Layer aufheben. Die Instance bleibt jedoch für den Stack registriert, sodass Sie diese ggf. einem anderen Layer zuweisen können.

Um die Zuweisung einer registrierten Instanz mithilfe der Konsole aufzuheben

1. Wählen Sie im Navigationsbereich Instances aus.
2. Wählen Sie die Instanz aus, deren Zuweisung Sie aufheben möchten.
3. Wählen Sie auf der Detailseite für die Instance die Option Zuweisung aufheben aus.



Um die Zuweisung einer registrierten Instanz aufzuheben, verwenden Sie AWS CLI

Führen Sie den aws opsworks unassign-instanceBefehl aus, um die Zuweisung einer registrierten Instanz zu allen Layern aufzuheben, die die Instanz verwenden.

```
aws opsworks unassign-instance --region region --instance-id instance-id
```

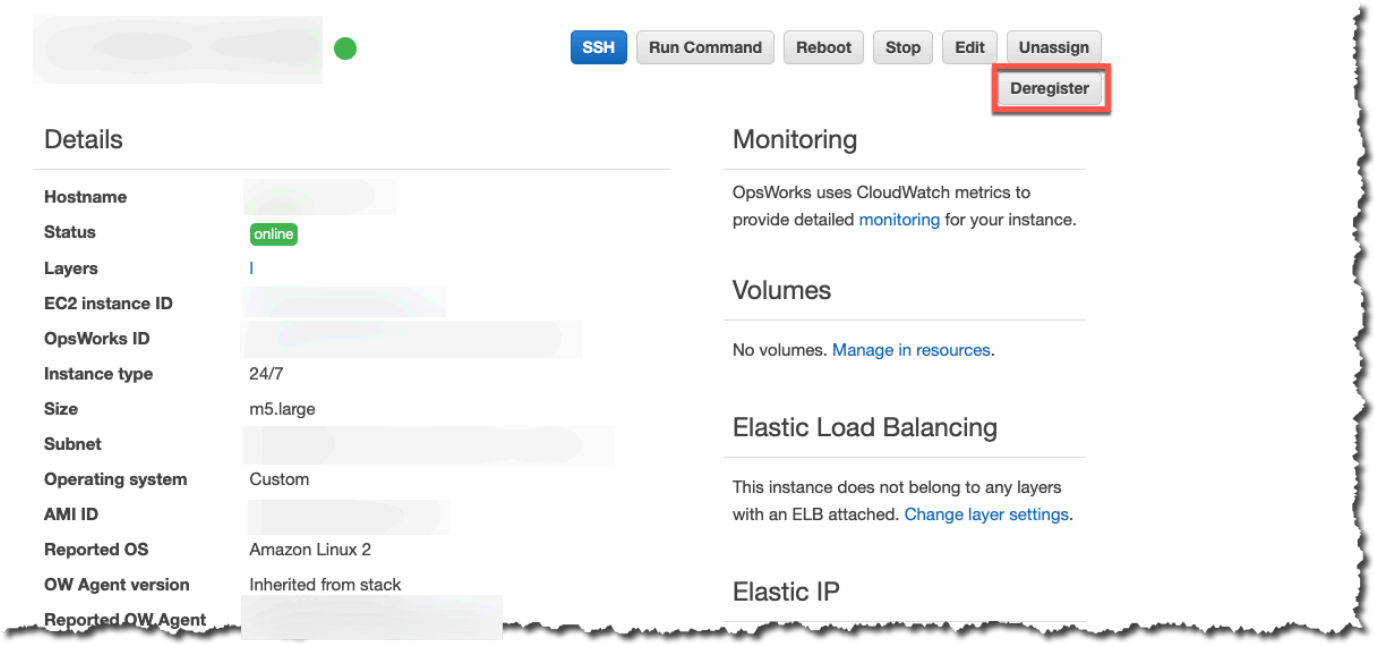
Aufheben einer Instance-Registrierung

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können die Registrierung einer Instanz über die AWS OpsWorks Konsole oder den SDK-Vorgang AWS CLI aufheben.

Um die Registrierung einer Instanz mithilfe der Konsole aufzuheben

1. Wählen Sie im Navigationsbereich Instances aus.
2. Wählen Sie die Instanz aus, deren Registrierung Sie aufheben möchten.
3. Wählen Sie auf der Detailseite für die Instance die Option Deregister aus.



Um die Registrierung einer Instance aufzuheben, verwenden Sie AWS CLI

Führen Sie den aws opsworks deregister-instanceBefehl aus, um eine Instanz von ihrem Stack abzumelden.

```
aws opsworks deregister-instance --region region --instance-id instance-id
```

Wenn Sie eine Instance deregistrieren, geht AWS OpsWorks Stacks wie folgt vor:

- Die Instance wird aus dem Stack entfernt.
- Die Zuweisung der Instance zu allen zugewiesenen Layern wird aufgehoben.
- Der Agent wird heruntergefahren und deinstalliert.
- Deregistriert alle angehängten Ressourcen (Elastic IP-Adressen und Amazon EBS-Volumes).

Dieses Verfahren umfasst Ressourcen, die vor der Registrierung an die Instance angehängt wurden, und Ressourcen, die Sie mithilfe von AWS OpsWorks Stacks an die Instance angehängt haben, als sie Teil des Stacks war. Nach der Aufhebung der Registrierung sind diese Ressourcen keine Stack-Ressourcen mehr, bleiben jedoch der Instance zuordnet.

- Bei lokalen Instances wird die Gebührenerhebung gestoppt.
- Entfernt alle Tags, die der Instanz OpsWorks hinzugefügt wurden.

Die Instanz bleibt im laufenden Zustand, steht jedoch unter Ihrer direkten Kontrolle und wird nicht mehr von AWS OpsWorks Stacks verwaltet.

> **Note**
>
> Sowohl das Registrieren als auch das Abmelden von Computern oder Instanzen werden nur innerhalb von Linux-Stacks vollständig unterstützt. Bei Windows-Stacks ist das Aufheben der Registrierung von Instanzen zulässig, der Agent wird dadurch jedoch nicht von der Instanz deinstalliert. OpsWorks Bei der Aufhebung der Registrierung werden nicht alle geänderten Dateien entfernt und es erfolgt auch keine vollständige Wiederherstellung mithilfe der Sicherungskopien bestimmter Dateien. Diese Liste gilt für Chef 11.10- und Chef 12-Stacks, die Unterschiede zwischen den beiden Versionen finden Sie hier.
>
> - Die Sicherung von `/etc/hosts` wird als `/var/lib/aws/opsworks/local-mode-cache/backup/etc/` gespeichert, aber nicht wiederhergestellt.
> - Einträge für `aws` und `opsworks` verbleiben in "passwd"-, "group"- und "shadow"-Dateien usw.
> - `/etc/sudoers`enthält einen Verweis auf ein Stacks-Verzeichnis AWS OpsWorks .
> - Die folgenden Dateien können bleiben, langfristig sollte `/var/lib/aws/opsworks` gelöscht werden.
>   - `/var/log/aws/opsworks` bleibt auf Instances in Chef 11.10-Stacks bestehen.
>   - `/var/lib/aws/opsworks` bleibt in Chef 11.10- und Chef 12-Stacks bestehen.
>   - `/var/chef` bleibt auf Instances in Chef 12-Stacks bestehen.
> - Weitere verbleibende Dateien sind:
>   - `/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`

Lebenszyklus einer registrierten Instance

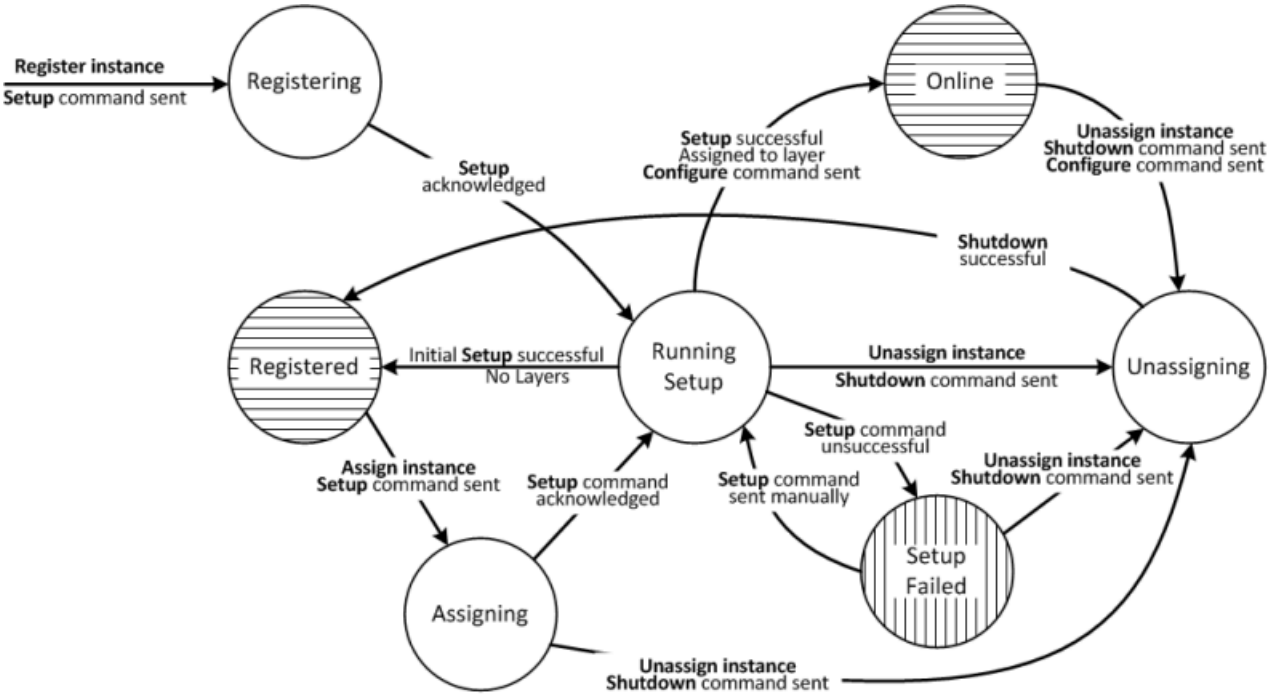
Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Diese Funktion wird nur für Linux-Stacks unterstützt.

Der Lebenszyklus einer registrierten Instance beginnt, wenn der Agent installiert ist und ausgeführt wird. An diesem Punkt wird AWS OpsWorks Stacks angewiesen, die Instanz beim Stack zu registrieren. Das folgende Statusdiagramm bietet eine Übersicht über die wichtigsten Elemente des Lebenszyklus.



Jeder Status entspricht einem Instance-Zustand. Die Kanten stehen für einen der folgenden AWS OpsWorks Stacks-Befehle. Details dazu finden Sie in den folgenden Abschnitten.

- Setup — Dieser Befehl entspricht dem Setup-Lifecycle-Ereignis und führt die Setup-Rezepte der Instanz aus.
- Configure — Dieser Befehl entspricht dem Configure Lifecycle-Ereignis.

  AWS OpsWorks Stacks löst dieses Ereignis bei jeder Instance im Stack aus, wenn eine Instance in den Online-Status wechselt oder diesen verlässt. Die Instances führen die Konfigurationsrezepte aus, sodass die für die Einbindung der neuen Instance erforderlichen Änderungen vorgenommen werden.
- Shutdown — Dieser Befehl entspricht dem Shutdown-Lifecycle-Ereignis, das die Shutdown-Rezepte der Instance ausführt.

  Diese Rezepte führen Aufgaben (wie z. B. Services herunterfahren) aus, stoppen jedoch nicht die Instance.
- Deregister — Dieser Befehl hebt die Registrierung einer Instance auf und entspricht keinem Lebenszyklusereignis.

> **Note**
>
> Aus Gründen der Übersichtlichkeit werden die Status "Deregistering" und "Deleted" im Diagramm nicht abgebildet. Sie können die Registrierung einer Instance in jedem Status des Diagramms aufheben. Dann wird der Befehl "Deregister" an die Instance übermittelt und diese wechselt in den Status "Deregistering".
>
> - Wenn Sie die Registrierung einer Online-Instance aufheben, sendet AWS OpsWorks Stacks einen Configure-Befehl an die verbleibenden Instances im Stack, um sie darüber zu informieren, dass die Instance offline geht.
> - Nach Ausführung des Befehls "Deregister" wird die Instance zwar weiter ausgeführt, befindet sich jedoch im Status "Delete" und ist nicht mehr Teil des Stacks. Soll die Instance wieder in den Stack aufgenommen werden, muss sie erneut registriert werden.

Themen

Registrieren

Nachdem der Agent eine Registrierungsanfrage gesendet hat, startet AWS OpsWorks Stacks den Instanzlebenszyklus, indem ein Setup-Befehl an die Instanz gesendet wird, wodurch sie in den Status Registrierung versetzt wird. Hat die Instance den Befehl "Setup" ausgeführt, ändert sich ihr Status in Status "Running Setup".

Status "Running Setup"

Im Status "Running Setup" werden die Einrichtungsrezepte für die Instance ausgeführt. Setup funktioniert abhängig vom vorherigen Status.

> **Note**
>
> Wenn Sie die Zuweisung der Instanz aufheben, während sie sich im Status Running Setup befindet, sendet AWS OpsWorks Stacks einen Shutdown-Befehl, der die Shutdown-Rezepte der Instanz ausführt, die Instanz jedoch nicht stoppt. Die Instance wechselt in den Status Status "Unassigning".

Themen

- Registrieren
- Status "Assigning"
- Status "Setup Failed"

Registrieren

Während des Registrierungsvorgangs erstellt das Setup eine AWS OpsWorks Stacks-Instanz, die die registrierte Instanz im Stack darstellt, und führt eine Reihe von grundlegenden Setup-Rezepten auf der Instanz aus.

Eine wichtige Änderung der Ersteinrichtung besteht im Überschreiben der Instance-Hosts-Datei. Durch die Registrierung der Instance haben Sie die Benutzerverwaltung an AWS OpsWorks Stacks übergeben, das für die Überprüfung der SSH-Anmeldeberechtigungen eine eigene Hosts-Datei benötigt. Bei der Ersteinrichtung werden zudem zahlreiche Dateien erstellt oder geändert, bei Ubuntu-Systemen werden auch Paketquellen geändert und mehrere Pakete installiert. Details hierzu finden Sie unter Konfigurationsänderungen im Rahmen der Ersteinrichtung.

Während der Registrierung ruft der Prozess das IAM auf`AttachUserPolicy`, das Teil der Berechtigungen ist, die dem IAM-Benutzer zugewiesen sind, den Sie als Voraussetzung erstellen. Wenn `AttachUserPolicy` nicht vorhanden ist (höchstwahrscheinlich, weil Sie eine ältere Version der AWS CLI ausführen), wird im Prozess stattdessen `PutUserPolicy` aufgerufen.

> **Note**
>
> Aus Konsistenzgründen führt AWS OpsWorks Stacks jedes zentrale Setup-Rezept aus. Bei einigen werden jedoch nur einige oder alle Aufgaben ausgeführt, sofern eine Instance mindestens einem Layer zugewiesen wurde, das heißt, die Ersteinrichtung ist nicht zwangsläufig betroffen.

- Bei erfolgreicher Einrichtung wechselt die Instance in den Status Status "Registered".
- Bei fehlerhafter Einrichtung wechselt die Instance in den Status Status "Setup Failed".

Status "Assigning"

Der Instanz ist mindestens eine Ebene zugewiesen. AWS OpsWorks Stacks führt die Setup-Rezepte jeder Ebene aus, einschließlich aller benutzerdefinierten Rezepte, die Sie dem Setup-Ereignis der Ebene zugewiesen haben.

- Bei erfolgreicher Einrichtung wechselt die Instance in den Status "Online" und AWS OpsWorks Stacks löst auf jeder Instance im Stack ein Configure-Lebenszyklusereignis aus, um diese über die neue Instance zu informieren.
- Schlägt die Einrichtung hingegen fehl, wechselt die Instance in den Status "Setup Failed".

> **Note**
>
> Im Rahmen dieser Einrichtung werden die Core-Rezepte ein zweites Mal ausgeführt. Chef-Rezepte sind jedoch idempotent, daher führen sie bereits ausgeführte Aufgaben nicht erneut aus.

Status "Setup Failed"

Falls die Einrichtung einer Instance im Status Status "Assigning" fehlschlägt, können Sie die Einrichtungsrezepte für die Instance mit dem Stack-Befehl "Setup" erneut manuell ausführen.

- Bei erfolgreicher Einrichtung wechselt die zugewiesene Instance in den Status Status "Online" und AWS OpsWorks Stacks löst auf jeder Instance im Stack ein Configure-Lebenszyklusereignis aus, um diese über die neue Instance zu informieren.
- Schlägt die Einrichtung fehl, wechselt die Instance wieder in den Status "Setup Failed".

Status "Registered"

Instanzen im Status Registriert sind Teil des Stacks und werden von AWS OpsWorks Stacks verwaltet, aber keiner Ebene zugewiesen. In diesem Status können sie unbegrenzt verweilen.

Wenn Sie die Instanz einer oder mehreren Ebenen zuweisen, sendet AWS OpsWorks Stacks einen Setup-Befehl an die Instanz und sie wechselt in den Status "Assigning" Status.

Status "Assigning"

Hat die Instance den Befehl "Setup" ausgeführt, ändert sich ihr Status in Status "Running Setup".

Wenn Sie die Zuweisung der Instanz aufheben, während sie sich im Status Zuweisen befindet, beendet AWS OpsWorks Stacks den Einrichtungsvorgang und sendet einen Shutdown-Befehl. Die Instance wechselt in den Status Status "Unassigning".

Status "Online"

Die Instance ist nun mindestens einem Layer zugewiesen und wird wie eine reguläre AWS OpsWorks Stacks-Instance behandelt. In diesem Status kann sie unbegrenzt verweilen.

Wenn Sie die Zuweisung der Instanz aufheben, während sie sich im Status Online befindet, sendet AWS OpsWorks Stacks einen Shutdown-Befehl an die Instance und einen Configure-Befehl an die restlichen Instanzen des Stacks. Die Instance wechselt in den Status Status "Unassigning".

Status "Setup Failed"

Der Befehl "Setup" konnte nicht ausgeführt werden.

- Sie können es erneut mit dem Stack-Befehl "Setup" versuchen.

  Die Instance kehrt in den Status Status "Running Setup" zurück.
- Wenn Sie die Zuweisung der Instance aufheben, sendet AWS OpsWorks Stacks einen Shutdown-Befehl an die Instance.

  Die Instance wechselt in den Status Status "Unassigning".

Status "Unassigning"

Nach Ausführung des Befehls "Shutdown" ist die Instance keinem Layer mehr zugeordnet und kehrt in den Status Status "Registered" zurück.

> **Note**
>
> Falls die Instance mehreren Layern zugewiesen ist, wird die Zuweisung für jeden Layer aufgehoben. Sie können die Zuweisung nicht nur für einige der zugewiesenen Layer aufheben. Wenn Sie andere Layer zuweisen möchten, heben Sie zunächst die Zuweisung der Instance auf und weisen anschließend die gewünschten Layer wieder zu.

Konfigurationsänderungen im Rahmen der Ersteinrichtung

Bei der Ersteinrichtung werden die folgenden Dateien und Verzeichnisse auf allen registrierten Instances erstellt oder geändert.

Erstellte Dateien

```
/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/
```

Geänderte Dateien

```
/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
```

Bei der Ersteinrichtung wird auch eine Swap-Datei auf Amazon EC2-Micro-Instances erstellt.

Folgende Änderungen werden im Rahmen der Ersteinrichtung für Ubuntu-Systeme ausgeführt.

Paketquellen

Die Paketquellen werden bei der Ersteinrichtung folgendermaßen geändert:

- `deb http://archive.ubuntu.com/ubuntu/ ${code_name} main universe`

  Bis: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name} main universe`
- `deb http://archive.ubuntu.com/ubuntu/ ${code_name}-updates main universe`

  Bis: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name}-updates main universe`
- `deb http://archive.ubuntu.com/ubuntu ${code_name}-security main universe`

  Bis: `deb-src http://archive.ubuntu.com/ubuntu ${code_name}-security main universe`
- `deb http://archive.ubuntu.com/ubuntu/ ${code_name}-updates multiverse`

  Bis: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name}-updates multiverse`
- `deb http://archive.ubuntu.com/ubuntu ${code_name}-security multiverse`

  Bis: `deb-src http://archive.ubuntu.com/ubuntu ${code_name}-security multiverse`
- `deb http://archive.ubuntu.com/ubuntu/ ${code_name} multiverse`

  Bis: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name} multiverse`
- `deb http://security.ubuntu.com/ubuntu ${code_name}-security multiverse`

  Bis: `deb-src http://security.ubuntu.com/ubuntu ${code_name}-security multiverse`

Pakete

Im Rahmen der Ersteinrichtung wird `landscape` deinstalliert und die folgenden Pakete werden installiert.

`autofs`	`libicu-dev`	`libopenssl-ruby`
`libssl-dev`	`libxml2-dev`	`libxslt-dev`

libyaml-dev	monit	ntpd
procps	ruby	ruby-dev
rubygems	screen	sqlite
vim	xfs	

Bearbeiten der Instance-Konfiguration

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können Instance-Konfigurationen, einschließlich registrierter Amazon Elastic Compute Cloud (Amazon EC2) -Instances, mit den folgenden Einschränkungen bearbeiten:

- Die Instances muss angehalten werden.

  Es ist für Online-Instances zwar nicht möglich, die Eigenschaften zu bearbeiten, Sie können durch bearbeiten der Instance-Layers jedoch einige Aspekte der Konfiguration beeinflussen. Weitere Informationen finden Sie unter Bearbeiten der Konfiguration einer Ebene OpsWorks .
- Einige Einstellungen wie Availability Zone und Scaling Type werden beim Erstellen der Instance festgelegt und können später nicht mehr geändert werden.
- Einige Einstellungen können nur für Instance Store-Backed Instances geändert werden, nicht für Amazon Elastic Block Store-gestützte Instances.

  Sie können beispielsweise das Betriebssystem einer Instance, die durch einen Store gesichert wird, ändern. Amazon EBS-gestützte Instances müssen das Betriebssystem verwenden, das Sie bei der Erstellung der Instance angegeben haben. Weitere Informationen zum Instance-Speicher finden Sie unter Storage.
- Standardmäßig erben Instances die Einstellung für die Agent-Version des Stacks.

Sie können die OpsWorks Agentenversion verwenden, um die Agentenversionseinstellung des Stacks zu überschreiben und eine bestimmte Agentenversion für eine Instance anzugeben. Wenn Sie die Agentenversion einer Instanz angeben, aktualisiert AWS OpsWorks Stacks den Agenten nicht automatisch, wenn eine neue Version verfügbar ist, auch wenn die Agentenversion des Stacks auf Automatisches Update eingestellt ist. Sie müssen die Agentenversion der Instanz manuell aktualisieren, indem Sie die Instanzkonfiguration bearbeiten. AWS OpsWorks Stacks installiert dann die angegebene Agentenversion auf der Instanz.

> **Note**
>
> Sie können die Konfiguration von registrierten lokalen Instances nicht bearbeiten.

So bearbeiten Sie die Instance-Konfiguration

1. Halten Sie die Instance an, falls Sie das noch nicht getan haben.
2. Klicken Sie auf der Seite Instances auf den Namen einer Instance, um die Seite Details anzuzeigen.
3. Klicken Sie auf Edit, um die Bearbeitungsseite anzuzeigen.
4. Bearbeiten Sie die Instance-Konfiguration.

Eine Beschreibung der Einstellungen Host name, Size, SSH key und Operating system finden Sie unter Hinzufügen einer Instance zu einem Layer. Mit der Einstellung Layers können Sie Layer hinzufügen oder entfernen. Die aktuellen Layer der Instance werden vor der Liste aller Layer angezeigt.

- Um einen Layer hinzuzufügen, wählen Sie ihn aus der Liste aus.
- Um die Instance aus einem Layer zu entfernen, klicken Sie auf das x neben dem entsprechenden Layer.

  Eine Instance muss zu mindestens einem Layer gehören. Der letzte Layer lässt sich daher nicht entfernen.

Wenn Sie die Instance neu starten, startet AWS OpsWorks Stacks eine neue Amazon EC2 EC2-Instance mit der aktualisierten Konfiguration.

AWS OpsWorks Stacks-Instances löschen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können AWS OpsWorks Stacks verwenden, um eine Instance zu stoppen, einschließlich registrierter Amazon EC2 EC2-Instances. Auf diese Weise wird die EC2-Instance gestoppt, verbleibt jedoch im Stack. Sie können sie neu starten, indem Sie auf start in der Spalte Actions (Aktionen) der Instance klicken. Wenn Sie eine Instance nicht mehr benötigen und sie aus dem Stack entfernen möchten, können Sie sie löschen. Dadurch wird die Instance aus dem Stack entfernt und die zugehörige Amazon EC2 EC2-Instance beendet. Durch das Löschen einer Instance werden auch alle zugehörigen Protokolle oder Daten sowie alle Amazon Elastic Block Store (EBS) -Volumes auf der Instance gelöscht.

> **Important**
>
> Dieses Thema gilt nur für Amazon EC2 EC2-Instances, die von AWS OpsWorks Stacks verwaltet werden. Weitere Informationen zum Löschen von Instances, die von der Amazon EC2 EC2-Konsole oder API verwaltet werden, finden Sie unter Terminate Your Instance.

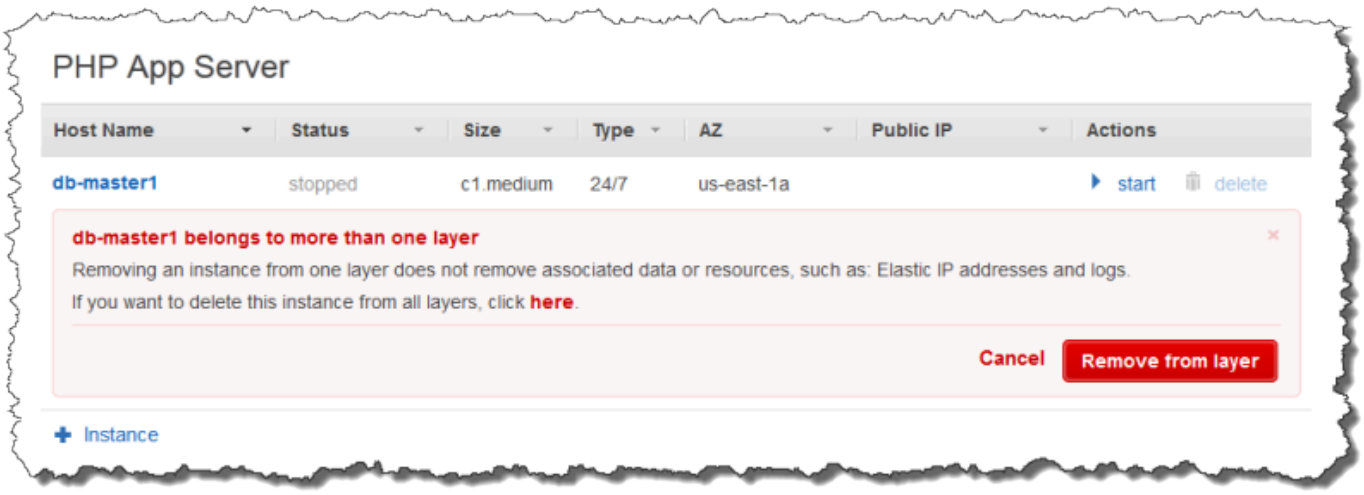
> **Note**
>
> Sie können AWS OpsWorks Stacks nicht verwenden, um eine registrierte lokale Instance zu löschen.

Falls eine Instance mehreren Layern angehört, können Sie die Instance aus dem Stack löschen oder nur einen bestimmten Layer entfernen. Sie können die Instance-Layer auch über die Instance-Konfiguration löschen, wie im Thema Bearbeiten der Instance-Konfiguration beschrieben.

> **Important**
>
> Sie sollten AWS OpsWorks Stacks-Instanzen nur mithilfe der Stacks-Konsole oder der AWS OpsWorks Stacks-API löschen. Insbesondere sollten Sie AWS OpsWorks Stacks-Instances nicht mithilfe der Amazon EC2 EC2-Konsole oder API löschen, da Amazon EC2 EC2-Aktionen nicht automatisch mit Stacks synchronisiert werden. AWS OpsWorks Wenn beispielsweise Auto Healing aktiviert ist und Sie eine Instance mithilfe der Amazon EC2 EC2-Konsole beenden, behandelt AWS OpsWorks Stacks die beendete Instance als ausgefallene Instance und startet eine weitere Amazon EC2 EC2-Instance, um sie zu ersetzen. Weitere Informationen finden Sie unter Verwenden von Auto Healing.

So löschen Sie eine Instance

1. Suchen Sie auf der Seite Instances unter dem entsprechenden Layer nach der Instance. Wenn die Instance ausgeführt wird, klicken Sie auf stop in der Spalte Actions.
2. Nachdem der Status sich in stopped geändert hat, klicken Sie auf delete. Wenn die Instance Mitglied von mehr als einer Ebene ist, zeigt Layer AWS OpsWorks Stacks den folgenden Abschnitt an.



   - Wenn die Instance nur aus dem ausgewählten Layer entfernt werden soll, klicken Sie auf Remove from layer.

     Die Instance verbleibt dann auf den anderen Layern und kann neu gestartet werden.
   - Um eine Instance aus allen Layern zu löschen und somit aus dem Stack zu entfernen, klicken Sie hier.

3. Wenn Sie sich dafür entscheiden, eine Instanz vollständig aus dem Stapel zu entfernen, oder wenn die Instanz nur Mitglied einer Ebene ist, werden Sie von AWS OpsWorks Stacks aufgefordert, das Löschen zu bestätigen.

   Wählen Sie zur Bestätigung Delete. Neben der Instanz aus dem Stack werden mit dieser Aktion alle zugeordneten Protokolle oder Daten sowie Stamm-Volumes gelöscht, die der Instance angefügt wurden. Um alle Instance-Volumes zu entfernen, wählen Sie die Option Delete instance's EBS volumes (snapshots will not be deleted) (EBS-Volumes der Instance löschen (Snapshots werden nicht gelöscht)), bevor Sie Delete (Löschen) auswählen.

Verwenden von SSH zum Anmelden bei einer Linux-Instance

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können sich mit SSH entweder über den integrierten MindTerm Client oder über einen Drittanbieter-Client wie PuTTY bei Ihren Online-Linux-Instances anmelden. SSH benötigt zur Authentifizierung in der Regel ein RSA-Schlüsselpaar. Sie installieren den öffentlichen Schlüssel auf der Instanz und stellen dem SSH-Client den entsprechenden privaten Schlüssel zur Verfügung. AWS OpsWorks Stacks übernimmt die Installation von öffentlichen Schlüsseln auf den Instances Ihres Stacks für Sie wie folgt.

- Amazon Elastic Compute Cloud (Amazon EC2) -Schlüsselpaar — Wenn die Region des Stacks über ein oder mehrere Amazon EC2 EC2-Schlüsselpaare verfügt, können Sie ein Standard-SSH-Schlüsselpaar für den Stack angeben.

  Im Zuge der Erstellung einer Instance können Sie optional das Standard-Schlüsselpaar überschreiben und ein anderes Paar definieren. In beiden Fällen installiert AWS OpsWorks Stacks den öffentlichen Schlüssel des angegebenen Schlüsselpaars auf der Instance. Weitere Informationen zum Erstellen von Amazon EC2 EC2-Schlüsselpaaren finden Sie unter Amazon EC2 EC2-Schlüsselpaare.

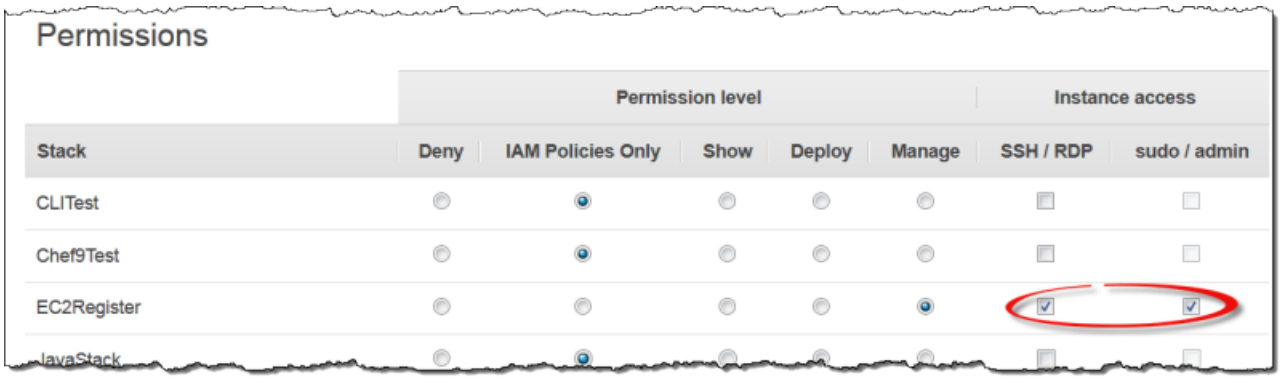
- Persönliches key pair — Jeder Benutzer kann ein persönliches key pair bei AWS OpsWorks Stacks registrieren.

  Der Benutzer oder ein Administrator registriert den öffentlichen Schlüssel bei AWS OpsWorks Stacks, und der Benutzer speichert den privaten Schlüssel lokal. Wenn Sie Stack-Berechtigungen festlegen, bestimmt der Administrator, welche Benutzer SSH-Zugriff auf die Stack-Instances haben. AWS OpsWorks Stacks erstellt automatisch für jeden autorisierten Benutzer einen Systembenutzer auf den Instanzen des Stacks und installiert den persönlichen öffentlichen Schlüssel des Benutzers.

Ein Benutzer muss über eine SSH-Autorisierung verfügen, um den MindTerm SSH-Client zu verwenden oder sein persönliches key pair zu verwenden, um sich bei den Instances eines Stacks anzumelden.

Um SSH für einen Benutzer zu autorisieren

1. Klicken Sie im AWS OpsWorks Stacks-Navigationsbereich auf Berechtigungen.
2. Wählen Sie SSH/RDP für den gewünschten IAM-Benutzer aus, um die erforderlichen Berechtigungen zu gewähren. Wenn Sie dem Benutzer die Möglichkeit geben möchten, Berechtigungen zu erhöhen, z. B. **sudo** um CLI-Befehle für Agenten auszuführen, wählen Sie ebenfalls sudo/admin aus.



Weitere Informationen zur Verwendung von Stacks zur Verwaltung des SSH-Zugriffs finden Sie unter. AWS OpsWorks Verwalten des SSH-Zugriffs

Themen

- Verwenden des integrierten SSH-Clients MindTerm
- Verwenden eines SSH-Clients von einem Drittanbieter

Verwenden des integrierten SSH-Clients MindTerm

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Der einfachste Weg, sich bei einer Linux-Instanz anzumelden, ist die Verwendung des integrierten MindTerm SSH-Clients. Jede Online-Instanz enthält eine SSH-Aktion, mit der Sie den MindTerm Client starten können.

> **Note**
>
> Sie müssen Java in Ihrem Browser aktiviert haben, um den MindTerm Client verwenden zu können.

Um sich mit dem MindTerm Client anzumelden

1. Wenn nicht bereits erfolgt, autorisieren Sie den SSH-Zugriff für den IAM-Benutzer, der eine Verbindung mit der Instance herstellen möchte, wie im vorherigen Abschnitt beschrieben.
2. Melden Sie sich als Benutzer an.
3. Wählen Sie auf der Seite Instances die Option ssh in der Spalte Actions für die entsprechende Instance aus.

4. Geben Sie für Private Key einen Pfad zum persönlichen privaten Schlüssel des Benutzers oder zu einem privaten Amazon EC2 EC2-Schlüssel an, je nachdem, welche öffentlichen Schlüssel Sie auf der Instance installiert haben.
5. Wählen Sie Launch Mindterm aus und verwenden Sie das Terminalfenster, um Befehle in der Instance auszuführen.

Verwenden eines SSH-Clients von einem Drittanbieter

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können auch einen SSH-Client eines Drittanbieters, wie z. B. PuTTY, zum Herstellen einer Verbindung mit Linux-Instances verwenden.

So verwenden Sie einen SSH-Client eines Drittanbieters

1. Stellen Sie sicher, dass AWS OpsWorks Stacks einen öffentlichen Amazon EC2 EC2-Schlüssel oder den persönlichen öffentlichen Schlüssel eines IAM-Benutzers auf der Instance installiert hat, wie bereits beschrieben.
2. Entnehmen Sie aus der Detailseite den öffentlichen DNS-Namen oder die öffentliche IP-Adresse der Instance.
3. Geben Sie den vom Betriebssystem abhängigen Client-Hostnamen wie folgt an:

   - Amazon Linux und Red Hat Enterprise Linux (RHEL) — `ec2-user@`*`DNSName/Address`*`.`
   - Ubuntu —`ubuntu@`*`DNSName/Address`*.

   Ersetzen Sie *`DNSName/Address`* mit dem öffentlichen DNS-Namen oder der IP-Adresse aus dem vorherigen Schritt.
4. Geben Sie dem Client den privaten Schlüssel an, der zu einem installierten öffentlichen Schlüssel passt. Sie können entweder einen privaten Amazon EC2 EC2-Schlüssel oder

den persönlichen privaten Schlüssel eines IAM-Benutzers verwenden, je nachdem, welche öffentlichen Schlüssel auf der Instance installiert wurden.

Verwenden von RDP zum Anmelden bei einer Windows-Instance

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können sich unter Verwendung des Windows Remote Desktop Protocol (RDP) folgendermaßen bei einer Online-Windows-Instance anmelden:

- Die Instance muss über eine Sicherheitsgruppe mit einer Regel für eingehenden Datenverkehr verfügen, die den RDP-Zugriff gestattet.

  Weitere Informationen zu Sicherheitsgruppen finden Sie unter Verwenden von Sicherheitsgruppen.
- Normale Benutzer — AWS OpsWorks Stacks stellt autorisierten normalen Benutzern ein RDP-Passwort zur Verfügung, das für einen begrenzten Zeitraum gültig ist, der zwischen 30 Minuten und 12 Stunden liegen kann.

  Benutzer müssen nicht nur autorisiert sein, sondern auch mindestens über die Berechtigungsstufe Anzeigen verfügen, oder ihre zugehörigen Richtlinien AWS Identity and Access Management (IAM) müssen die Aktion zulassen. `opsworks:GrantAccess`
- Administratoren — Sie können sich mit dem Administratorkennwort für eine unbegrenzte Zeit anmelden.

  Wie später beschrieben, können Sie, wenn Sie ein Amazon Elastic Compute Cloud (Amazon EC2) -Schlüsselpaar für die Instance angegeben haben, dieses zum Abrufen des Administratorkennworts verwenden.

> **Note**
>
> In diesem Thema wird beschrieben, wie Sie sich mit dem Windows Remote Desktop Connection Client von einer Windows-Workstation anmelden können. Sie können auch eine der verfügbaren RDP-Clients für Linux oder OS X verwenden, wobei dann das Verfahren etwas anders ist. Weitere Informationen über RDP-Clients, die mit Microsoft Windows Server 2012 R2 kompatibel sind, finden Sie unter Microsoft Remote Desktop Clients.

Themen

Bereitstellen einer Sicherheitsgruppe, die den RDP-Zugriff zulässt

Bevor Sie RDP zum Anmelden bei einer Windows-Instance verwenden können, müssen die Regeln für den eingehenden Datenverkehr der Sicherheitsgruppe der Instance RDP-Verbindungen zulassen. Wenn Sie den ersten Stack in einer Region erstellen, erstellt AWS OpsWorks Stacks eine Reihe von Sicherheitsgruppen. Dazu gehört auch eine mit dem Namen etwa`AWS-OpsWorks-RDP-Server`, die AWS OpsWorks Stacks an alle Windows-Instances anhängt, um den RDP-Zugriff zu ermöglichen. Standardmäßig sind in diesen Sicherheitsgruppe jedoch keine Regeln enthalten. Daher müssen Sie eine Regel für den eingehenden Datenverkehr zum Zulassen von RDP-Zugriff auf Ihre Instances hinzufügen.

So ermöglichen Sie den RDP-Zugriff

1. Öffnen Sie die Amazon EC2 EC2-Konsole, stellen Sie sie auf die Region des Stacks ein und wählen Sie im Navigationsbereich Sicherheitsgruppen aus.
2. Wählen Sie AWS- OpsWorks -RDP-Server, klicken Sie auf die Registerkarte Inbound und dann auf Bearbeiten.
3. Wählen Sie Add Rule (Regel hinzufügen) aus und legen Sie die folgenden Einstellungen fest:
   - Typ — RDP
   - Quelle — Die zulässigen Quell-IP-Adressen.

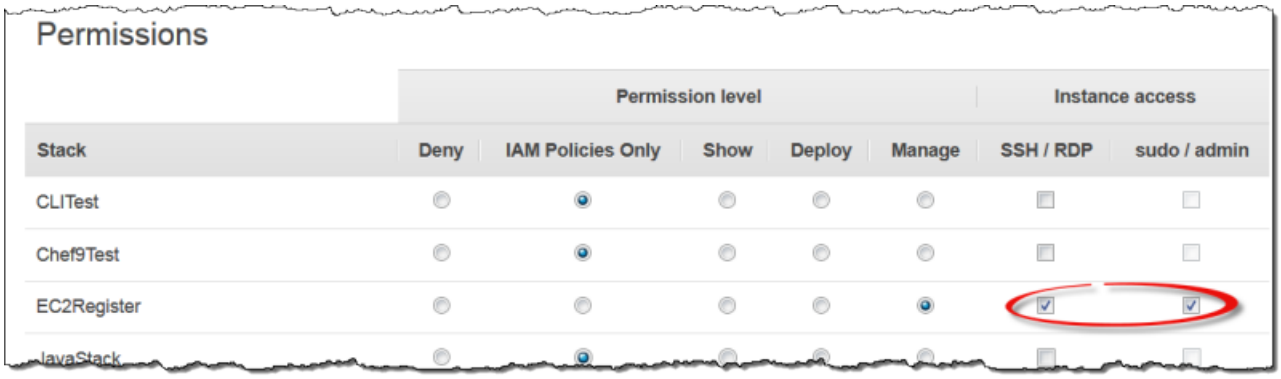
In der Regel erlauben Sie eingehende RDP-Anfragen von Ihrer eigenen IP-Adresse oder einem festen IP-Adressbereich (üblicherweise der IP-Adressbereich Ihres Unternehmens).

Anmelden als normaler Benutzer

Berechtigte Benutzer können sich mit einem von AWS OpsWorks Stacks bereitgestellten temporären Passwort bei Instances anmelden.

Um RDP für einen Benutzer zu autorisieren;

1. Klicken Sie im AWS OpsWorks Stacks-Navigationsbereich auf Berechtigungen.
2. Wählen Sie das SSH/RDP-Kontrollkästchen für den gewünschten Benutzer aus, um die erforderlichen Berechtigungen zu gewähren. Wenn Sie dem Benutzer außerdem auch Administratorberechtigungen gewähren möchten, wählen Sie noch sudo/admin aus.



Autorisierte Benutzer können sich bei einem der Online-Instances des Stacks folgendermaßen anmelden.

Um sich als normaler IAM-Benutzer anzumelden

1. Melden Sie sich als IAM-Benutzer an.
2. Wählen Sie auf der Seite Instances die Option rdp in der Spalte Actions für die entsprechende Instance aus.
3. Geben Sie die Sitzungsdauer an, die zwischen 30 Minuten und 12 Stunden betragen kann, und klicken Sie auf Generate Password. Das Passwort ist nur für die angegebene Sitzungsdauer gültig.
4. Notieren Sie die Werte für public DNS name, username und password und wählen Sie dann Acknowledge and close.

5. Öffnen Sie den Windows Remote Desktop Connection-Client, wählen Sie Show Options aus und geben Sie die folgenden, im Schritt 4 notierten Informationen an:

   - Computer — Der öffentliche DNS-Name der Instanz.

     Sie können alternativ auch die öffentliche IP-Adresse einfügen. Wählen Sie Instances aus und kopieren Sie Adresse aus der Spalte Public IP der Instance.

   - Benutzername — Der Benutzername.

6. Wenn die Client-Eingabeaufforderung für Ihre Anmeldeinformationen erscheint, geben Sie das Passwort ein, das Sie in Schritt 4 gespeichert haben.

> **Note**
>
> AWS OpsWorks Stacks generiert ein Benutzerkennwort nur für Online-Instanzen. Wenn Sie eine Instance starten und beispielsweise eines Ihrer benutzerdefinierten Einrichtungsrezepte fehlschlägt, geht die Instance in den Status `setup_failed` über. Auch wenn die Instance für AWS OpsWorks Stacks nicht online ist, läuft die EC2-Instance und es ist oft nützlich, sich anzumelden, um das Problem zu beheben. AWS OpsWorks Stacks generiert in diesem Fall kein Passwort für Sie, aber wenn Sie der Instance ein SSH-Schlüsselpaar zugewiesen haben, können Sie die EC2-Konsole oder CLI verwenden, um das Administratorkennwort der Instance abzurufen und sich als Administrator anzumelden. Weitere Informationen finden Sie im folgenden Abschnitt.

Anmelden als Administrator

Sie können sich bei einer Instance mit dem entsprechenden Passwort als Administrator anmelden. Wenn Sie einer Instance ein EC2-Schlüsselpaar zugewiesen haben, verwendet Amazon EC2 es, um beim Start der Instance automatisch ein Administratorkennwort zu erstellen und zu verschlüsseln. Anschließend können Sie mit dem Schlüsselpaar mithilfe der EC2-Konsole, einer API oder CLI das Passwort abrufen und entschlüsseln.

> **Note**
>
> Sie können kein persönliches SSH-Schlüsselpaar zum Abrufen eines Administratorpassworts verwenden. Sie müssen ein EC2-Schlüsselpaar benutzen.

Im Folgenden wird beschrieben, wie Sie mit der EC2-Konsole ein Administratorpasswort abrufen und sich bei einer Instance anmelden. Wenn Sie Befehlszeilen-Tools bevorzugen, können Sie auch den AWS CLI-Befehl `get-password-data` zum Abrufen des Passworts verwenden.

So melden Sie sich als Administrator an

1. Stellen Sie sicher, dass Sie ein EC2-Schlüsselpaar für die Instance angegeben haben. Sie können beim Erstellen des Stacks ein Standard-Schlüsselpaar für alle Stack-Instances angeben oder Sie können beim Erstellen des Stacks ein Schlüsselpaar für eine bestimmte Instance angeben.
2. Öffnen Sie die EC2-Konsole, wählen Sie erst die Region des Stacks und anschließend im Navigationsbereich Instances aus.
3. Wählen Sie zunächst die Instance, dann Connect und anschließend Get Password aus.
4. Geben Sie einen Pfad für das EC2-Schlüsselpaar auf Ihrer Workstation an und wählen Sie Decrypt Password aus. Kopieren Sie das entschlüsselte Passwort für eine spätere Nutzung.
5. Öffnen Sie den Windows Remote Desktop Connection-Client, wählen Sie Show Options aus und geben Sie die folgenden Informationen an:
   - Computer — Der öffentliche DNS-Name oder die öffentliche IP-Adresse der Instance, die Sie auf der Detailseite der Instance abrufen können.
   - Benutzername —`Administrator`.
6. Wenn die Client-Eingabeaufforderungen für Ihre Anmeldeinformationen erscheint, geben Sie das im Schritt 4 entschlüsselte Passwort ein.

Apps

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Eine AWS OpsWorks Stacks-App steht für Code, den Sie auf einem Anwendungsserver ausführen möchten. Der Code selbst befindet sich in einem Repository wie einem Amazon S3 S3-Archiv. Die App enthält die Informationen, die für die Bereitstellung des Codes auf den entsprechenden Anwendungsserver-Instances erforderlich sind.

Wenn Sie eine Anwendung bereitstellen, löst AWS OpsWorks Stacks ein Deploy-Ereignis aus, das die Deploy-Rezepte jeder Ebene ausführt. AWS OpsWorks Stacks installiert außerdem Stackkonfigurations- und Bereitstellungsattribute, die alle Informationen enthalten, die für die Bereitstellung der App erforderlich sind, z. B. das Repository der App und die Datenbankverbindungsdaten.

Sie müssen benutzerdefinierte Rezepte implementieren, die die Bereitstellungsdaten der App aus den Stack-Konfigurations- und Bereitstellungsattributen abrufen und die Bereitstellung durchführen.

Themen

Hinzufügen von Apps

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Im ersten Schritt bei der Bereitstellung einer Anwendung für Ihre Anwendungsserver fügen Sie eine App zum Stack hinzu. Die App stellt die Anwendung dar und enthält eine Vielzahl von Metadaten, wie z. B. den Namen und den Typ der Anwendung, sowie die Informationen, die zum Bereitstellen der Anwendung für die Server-Instances erforderlich sind, z. B. die Repository-URL. Sie müssen über die Manage-Berechtigungen verfügen, um eine App zu einem Stack hinzufügen zu können. Weitere Informationen finden Sie unter Verwalten von Benutzerberechtigungen.

> **Note**
>
> Das Verfahren in diesem Abschnitt gilt für Chef 12 und neuere Stacks. Weitere Informationen darüber, wie Anwendungen Ebenen in Chef 11-Stacks hinzugefügt werden, finden Sie unter Schritt 2.4: Erstellen und Bereitstellen einer Anwendung – Chef 11.

So fügen Sie eine App zu einem Stack hinzu

1. Platzieren Sie den Code in Ihrem bevorzugten Repository — einem Amazon S3 S3-Archiv, einem Git-Repository, einem Subversion-Repository oder einem HTTP-Archiv. Weitere Informationen finden Sie unter Anwendungsquelle.
2. Klicken Sie im Navigationsbereich auf Apps. Klicken Sie auf der Seite Apps auf Add an app (App hinzufügen) für Ihre erste App. Klicken Sie für alle nachfolgenden Apps auf +App.
3. Konfigurieren Sie die App auf der Seite App New (App neu) gemäß den Schritten im folgenden Abschnitt.

Konfigurieren einer App

Die Seite Add App (App hinzufügen) besteht aus den folgenden Abschnitten: Settings (Einstellungen), Application source (Anwendungsquelle), Data Sources (Datenquellen), Environment Variables (Umgebungsvariablen), Add Domains (Domänen hinzufügen) und SSL Settings (SSL-Einstellungen).

Themen

- Einstellungen
- Anwendungsquelle
- Datenquellen
- Umgebungsvariablen
- Domänen- und SSL-Einstellungen

Einstellungen

Name

Der Name der App, der verwendet wird, um die App in der Benutzeroberfläche darzustellen. AWS OpsWorks Stacks verwendet diesen Namen auch, um einen Kurznamen für die App zu generieren, der intern verwendet wird, und um die App in der Stackkonfiguration und den Bereitstellungsattributen zu identifizieren. Nach dem Hinzufügen der App zum Stack können Sie die Kurzbezeichnung anzeigen, indem Sie im Navigationsbereich auf Apps und dann auf den App-Namen klicken, um die Detailseite zu öffnen.

Document root (Basisverzeichnis)

AWS OpsWorks Stacks weist dem `[:document_root]`Attribut in den Attributen der App die Einstellung Document Root zu. `deploy` Der Standardwert ist `null`. Ihre Bereitstellungsrezepte können diesen Wert mithilfe der standardmäßigen Chef-Knotensyntax aus den `deploy`-Attributen abrufen und den angegebenen Code am entsprechenden Speicherort auf dem Server bereitstellen. Weitere Informationen zum Bereitstellen von Apps finden Sie unter Bereitstellungsrezepte.

Anwendungsquelle

Sie können Apps aus den folgenden Repository-Typen bereitstellen: Git, Amazon S3 S3-Bundle, HTTP-Bundle und Andere. Für alle Repository-Typen müssen Sie den Typ und die URL des Repositorys angeben. Einzelne Repository-Typen haben ihre eigenen Anforderungen, wie im Folgenden beschrieben.

> **Note**
>
> AWS OpsWorks Stacks stellt automatisch Anwendungen aus den Standard-Repositorys auf die integrierten Serverschichten bereit. Wenn Sie den Repository-Typ „Anderes Repository“ verwenden, was die einzige Option für Windows-Stacks ist, AWS OpsWorks fügt Stacks die Repository-Informationen in die `deploy`Attribute der App ein. Sie müssen jedoch benutzerdefinierte Rezepte implementieren, um die Bereitstellungsaufgaben zu erledigen.

Themen

- HTTP-Archiv
- Amazon S3 S3-Archiv

- Git-Repository
- Andere Repositorys

HTTP-Archiv

So verwenden Sie einen öffentlich verfügbaren HTTP-Server als Repository:

1. Erstellen Sie ein komprimiertes Archiv — zip, gzip, bzip2, Java WAR oder tarball — des Ordners, der den Code der App und alle zugehörigen Dateien enthält.

   > **Note**
   >
   > AWS OpsWorks Stacks unterstützt keine unkomprimierten Tarballs.

2. Laden Sie die Archivdatei auf den Server hoch.
3. Zum Festlegen des Repositorys in der Konsole wählen Sie das HTTP-Archiv als Repository-Typ aus und geben Sie die URL ein.

   Wenn das Archiv kennwortgeschützt ist, geben Sie unter Anwendungsquelle die Anmeldeinformationen an.

Amazon S3 S3-Archiv

So verwenden Sie einen Amazon Simple Storage Service-Bucket als Repository:

1. Erstellen Sie einen öffentlichen oder privaten Amazon S3 S3-Bucket. Weitere Informationen finden Sie in der Amazon S3 S3-Dokumentation.
2. Damit AWS OpsWorks Stacks auf private Buckets zugreifen kann, müssen Sie ein Benutzer mit mindestens Leserechten für den Amazon S3 S3-Bucket sein und Sie benötigen die Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel. Weitere Informationen finden Sie in der AWS Identity and Access Management Dokumentation.
3. Platzieren Sie den Code und alle zugehörigen Dateien in einem Ordner und speichern Sie diesen in einem komprimierten Archiv (ZIP, GZIP, BZIP2, Java WAR oder Tarball).

   > **Note**
   >
   > AWS OpsWorks Stacks unterstützt keine unkomprimierten Tarballs.

4. Laden Sie die Archivdatei in den Amazon S3 S3-Bucket hoch und notieren Sie die URL.
5. Um das Repository in der AWS OpsWorks Stacks-Konsole anzugeben, setzen Sie den Repository-Typ auf S3-Archiv und geben Sie die URL des Archivs ein. Für ein privates Archiv müssen Sie auch eine AWS-Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel angeben, dessen Richtlinie Berechtigungen für den Zugriff auf den Bucket gewährt. Lassen Sie diese Einstellungen für öffentliche Archive leer.

Git-Repository

Ein Git-Repository bietet Quellcodeverwaltung und Versionierung. AWS OpsWorks Stacks unterstützt öffentlich gehostete Repository-Sites wie GitHubBitbucket sowie privat gehostete Git-Server.
Bei Anwendungs- und Git-Submodulen hängt das Format, das Sie für die Repository-URL unter Application Source (Anwendungsquelle) festlegen, davon ab, ob das Repository öffentlich oder privat ist:

Öffentliches Repository — Verwenden Sie die schreibgeschützten HTTPS- oder Git-Protokolle. Erste Schritte mit Chef 11 Linux-StacksVerwendet beispielsweise ein öffentliches GitHub Repository, auf das über eines der folgenden URL-Formate zugegriffen werden kann:

- Schreibgeschütztes Git-Protokoll: **`git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git`**
- HTTPS: **`https://github.com/amazonwebservices/opsworks-demo-php-simple-app.git`**

Privates Repository — Verwenden Sie das SSH-Lese-/Schreibformat, das in diesen Beispielen gezeigt wird:

- Github-Repositorys: **`git@github.com:`*`project`*`/`*`repository`***.
- Repositorys auf einem Git-Server: ***`user`*`@`*`server`*`:`*`project`*`/`*`repository`***

Wenn Sie Git (Git) unter Source Control (Quellsteuerung) auswählen, werden zwei zusätzliche optionale Einstellungen angezeigt:

Repository SSH key (Repository-SSH-Schlüssel)

> Sie müssen einen SSH-Bereitstellungsschlüssel für den Zugriff auf private Git-Repositorys angeben. Dieses Feld benötigt den privaten Schlüssel; der öffentliche Schlüssel ist Ihrem Git-

Repository zugeordnet. Bei Git-Submodulen muss der angegebene Schlüssel Zugriff auf diese Submodule haben. Weitere Informationen finden Sie unter Verwenden von Git-Repository-SSH-Schlüsseln.

> **Important**
>
> Für den SSH-Bereitstellungsschlüssel ist kein Passwort erforderlich; AWS OpsWorks Stacks hat keine Möglichkeit, es weiterzugeben.

Branch/Revision

Wenn das Repository mehrere Branches hat, lädt AWS OpsWorks Stacks standardmäßig den Master-Branch herunter. Um einen bestimmten Branch festzulegen, geben Sie den Branch-Namen (SHA1-Hash) oder den Tag-Namen ein. Um einen bestimmten Commit festzulegen, geben Sie die vollständige Commit-ID mit 40 Hexadezimalziffern an.

Andere Repositorys

Wenn die Standard-Repositorys nicht Ihren Anforderungen entsprechen, können Sie andere Repositorys verwenden, z. B. Bazaar. AWS OpsWorks Stacks stellt Apps aus solchen Repositorys jedoch nicht automatisch bereit. Sie müssen benutzerdefinierte Rezepte zum Durchführen des Bereitstellungsverfahrens implementieren und diese den Bereitstellungsereignissen des entsprechenden Layers zuweisen. Ein Beispiel für die Implementierung von Bereitstellungsrezepten finden Sie unter Bereitstellungsrezepte.

Datenquellen

In diesem Abschnitt wird erläutert, wie Sie eine Datenbank an die Anwendung anfügen. Ihnen stehen folgende Optionen zur Verfügung:

- RDS — Hängen Sie eine der Amazon RDS-Serviceschichten des Stacks an.
- Keine — Hängen Sie keinen Datenbankserver an.

Wenn Sie RDS auswählen, müssen Sie Folgendes angeben.

Datenbank-Instance

Die Liste umfasst alle Amazon RDS-Serviceschichten. Sie können auch eine der folgenden Optionen auswählen:

(Erforderlich) Geben Sie an, welcher Datenbankserver an die Anwendung angefügt werden soll. Der Inhalt der Liste hängt von der Datenquelle ab.

- RDS — Eine Liste der Amazon RDS-Serviceschichten des Stacks.

Datenbankname

(Optional) Geben Sie einen Datenbanknamen an.

- Amazon RDS-Layer — Geben Sie den Datenbanknamen ein, den Sie für die Amazon RDS-Instance angegeben haben.

  Sie können den Datenbanknamen von der Amazon RDS-Konsole abrufen.

Wenn Sie eine App mit einer angehängten Datenbank bereitstellen, fügt AWS OpsWorks Stacks die Verbindung der Datenbank-Instance zu den `deploy`Attributen der App hinzu.

Sie können ein benutzerdefiniertes Rezept schreiben, um die Informationen aus den `deploy`-Attributen abzurufen, und dieses in die Datei einfügen, auf die die Anwendung zugreifen kann. Beim Anwendungstyp "Other" ist dies die einzige Option für die Bereitstellung der Datenbankverbindungsinformationen.

Weitere Informationen zum Verarbeiten von Datenbankverbindungen finden Sie unter Verbinden mit einer Datenbank.

Um einen Datenbankserver von einer Anwendung zu trennen, bearbeiten Sie die Anwendungskonfiguration und geben Sie einen anderen Datenbankserver oder keinen Server an.

Umgebungsvariablen

Sie können für jede Anwendung eine Reihe von Umgebungsvariablen definieren, die für die Anwendung spezifisch sind. Wenn Sie zum Beispiel über zwei Anwendungen verfügen, stehen die Umgebungsvariablen, die Sie für die erste Anwendung definieren, nicht für die zweite Anwendung zur Verfügung und umgekehrt. Sie können auch dieselbe Umgebungsvariable für mehrere Anwendungen definieren und jeder Anwendung einen anderen Wert zuweisen.

> **Note**
>
> Es gibt keinen besonderen Grenzwert in Bezug auf die Anzahl der Umgebungsvariablen. Die Größe der zugehörigen Datenstruktur, die die Namen, Werte und geschützten Flag-Werte der Variablen umfasst, darf jedoch 20 KB nicht überschreiten. Dieser Grenzwert sollte für die meisten, wenn nicht sogar für alle Anwendungsfälle ausreichend sein. Bei Überschreitung tritt ein Servicefehler (Konsole) oder eine Ausnahme (API) auf und es wird folgende Meldung angezeigt: "Environment: is too large (maximum is 20KB)."

AWS OpsWorks Stacks speichert die Variablen als Attribute in den Attributen der App. `deploy` Sie können diese Werte mithilfe der standardmäßigen Chef-Knotensyntax durch Ihre benutzerdefinierten Rezepte abrufen. Weitere Beispiele für den Zugriff auf die Umgebungsvariablen einer Anwendung finden Sie unter Verwenden von -Umgebungsvariablen.

Schlüssel

Der Name der Variable. Er kann bis zu 64 Groß- und Kleinbuchstaben, Zahlen und Unterstriche (_) enthalten, aber er muss mit einem Buchstaben oder Unterstrich beginnen.

Wert

Der Wert der Variable. Er kann bis zu 256 Zeichen enthalten, die alle druckbar sein müssen.

Geschützter Wert

Gibt an, ob der Wert geschützt ist. Diese Einstellung ermöglicht es Ihnen, sensible Daten wie Passwörter zu verbergen. Wenn Sie nach dem Erstellen der Anwendung den Wert Protected value (Geschützter Wert) für eine Variable festlegen:

- Auf der Detailseite der Anwendung wird nur der Name der Variable und nicht der Wert angezeigt.
- Wenn Sie über die Berechtigung zum Bearbeiten der Anwendung verfügen, können Sie auf Update value (Wert aktualisieren) klicken, um einen neuen Wert anzugeben. Sie können den alten Wert jedoch nicht anzeigen oder bearbeiten.

> **Note**
>
> Chef-Bereitstellungsprotokolle können manchmal auch Umgebungsvariablen enthalten. Dies bedeutet, dass geschützte Variablen möglicherweise in der Konsole angezeigt werden. Um

zu verhindern, dass geschützte Variablen in der Konsole angezeigt werden, empfehlen wir, Amazon S3 S3-Buckets als Speicher für geschützte Variablen zu verwenden, die nicht in der Konsole angezeigt werden sollen. Ein Beispiel dafür, wie Sie einen S3-Bucket für diesen Zweck verwenden finden Sie unter Verwenden eines Amazon S3 S3-Buckets in diesem Handbuch.

Domänen- und SSL-Einstellungen

Für den App-Typ Andere fügt AWS OpsWorks Stacks die Einstellungen zu den Attributen der `deploy` App hinzu. Ihre Rezepte können die Daten aus diesen Attributen abrufen und den Server gegebenenfalls konfigurieren.

Domäneneinstellungen

Dieser Abschnitt enthält das optionale Feld Add Domains (Domänen hinzufügen) für die Angabe von Domänen. Weitere Informationen finden Sie unter Verwenden von benutzerdefinierten Domänen.

SSL-Einstellungen

Dieser Abschnitt enthält den Schalter SSL Support (SSL-Support), mit dem Sie SSL aktivieren oder deaktivieren können. Wenn Sie auf Yes (Ja) klicken, müssen Sie SSL-Zertifikatinformationen angeben. Weitere Informationen finden Sie unter Verwenden von SSL.

Bereitstellen von Anwendungen

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

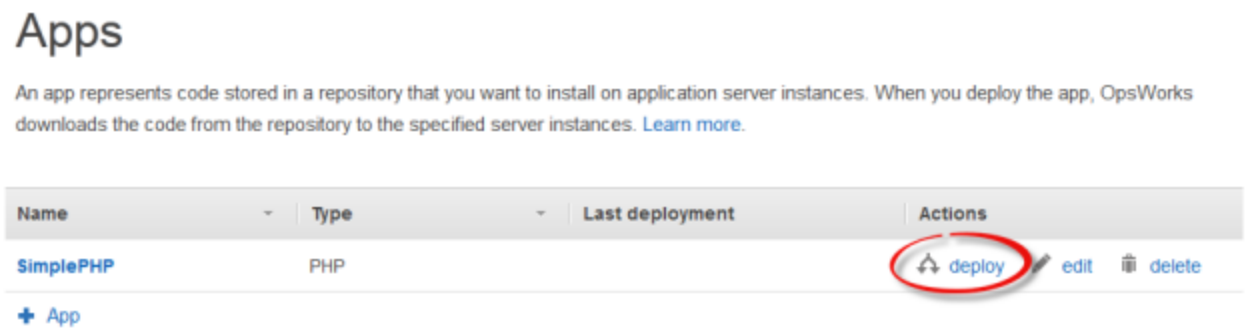
Der Hauptzweck der Bereitstellung besteht darin, auf Anwendungsserver-Instances Anwendungs-Code und verwandte Dateien bereitzustellen. Der Bereitstellungsvorgang wird von den

Bereitstellungsrezepten der jeweiligen Instance, die durch den Layer der Instance ermittelt werden, ausgeführt.

Wenn du eine Instanz startest, führt AWS OpsWorks Stacks nach Abschluss der Setup-Rezepte automatisch die Deploy-Rezepte der Instanz aus. Wenn Sie jedoch eine Anwendung hinzufügen oder sie ändern, müssen Sie diese auf jeder Online-Instance manuell bereitstellen. Zum Bereitstellen einer Anwendung müssen Sie über Berechtigungen zum Verwalten oder Bereitstellen verfügen. Weitere Informationen finden Sie unter Verwalten von Benutzerberechtigungen.

Bereitstellen einer Anwendung

1. Klicken Sie auf der Seite Apps auf die Aktion deploy (Bereitstellen) der App.



> **Note**
>
> Sie können eine Anwendung auch bereitstellen, indem Sie im Navigationsbereich auf Deployments (Bereitstellungen) klicken. Klicken Sie auf der Seite Deployments & Commands (Bereitstellungen und Befehle) auf Deploy an app (App bereitstellen). Dort können Sie auch auswählen, welche Anwendung Sie bereitstellen möchten.

2. Machen Sie folgende Angaben:

   - (Erforderlich) Stellen Sie Command: (Befehl:) auf deploy (Bereitstellen) ein, sofern diese Option noch nicht ausgewählt wurde.
   - (Optional) Geben Sie einen Kommentar ein.

3. Klicken Sie auf Erweitert >>, um ein benutzerdefiniertes JSON anzugeben. AWS OpsWorks Stacks fügt dem Knotenobjekt eine Reihe von Stackkonfigurations- und Bereitstellungsattributen hinzu. Die `deploy` Attribute enthalten die Bereitstellungsdetails und können von Bereitstellungsrezepten für Installations- und Konfigurationszwecke verwendet werden. Auf Linux-Stacks können Sie das benutzerdefinierte JSON-Feld verwenden, um die AWS OpsWorks

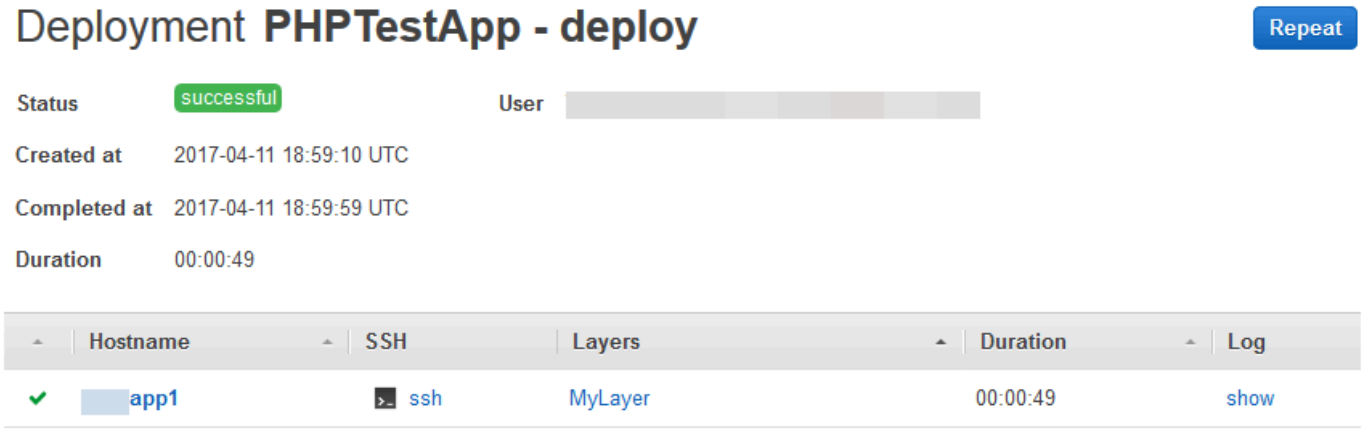
Standard-Stacks-Einstellungen zu überschreiben oder benutzerdefinierte Einstellungen an Ihre benutzerdefinierten Rezepte zu übergeben. Weitere Informationen zur Verwendung benutzerdefinierter JSON-Formate finden Sie unter Nutzen eines benutzerdefinierten JSON-Objekts.

> **Note**
>
> Wenn Sie hier ein benutzerdefiniertes JSON-Format angeben, wird es der Stack-Konfiguration und den Bereitstellungsattributen nur für diese Bereitstellung hinzugefügt. Wenn Sie dauerhaft ein benutzerdefiniertes JSON-Format hinzufügen möchten, müssen Sie es dem Stack hinzufügen. Ein benutzerdefiniertes JSON-Format ist auf 120 KB begrenzt. Wenn Sie mehr Kapazität benötigen, empfehlen wir, einige Daten auf Amazon S3 zu speichern. Ihre benutzerdefinierten Rezepte können dann die AWS-CLI oder das AWS SDK für Ruby verwenden, um Daten aus dem Bucket auf Ihre Instance herunterzuladen. Ein Beispiel finden Sie unter Verwenden des -SDK for Ruby.

4. Klicken Sie unter Instances auf Advanced >> (Erweitert >>) und geben Sie an, auf welchen Instances der Befehl zur Bereitstellung ausgeführt werden soll.

   Der Befehl "deploy" löst ein Bereitstellungsereignis aus, das die Bereitstellungsrezepte auf den ausgewählten Instances ausführt. Die Bereitstellungsrezepte für den zugehörigen Anwendungsserver laden den Code und die verwandten Dateien aus dem Repository herunter und installieren sie auf der Instance, sodass Sie in der Regel alle zugehörigen Anwendungsserver-Instances auswählen. Andere Instance-Typen benötigen jedoch unter Umständen Konfigurationsänderungen, um mit der neuen Anwendung umgehen zu können, daher ist es oft sinnvoll, auch auf diesen Instances Bereitstellungsrezepte auszuführen. Diese Rezepte aktualisieren die Konfiguration bei Bedarf, installieren jedoch nicht die Dateien der Anwendung. Weitere Informationen zu Rezepten finden Sie unter Cookbooks und Rezepte.

5. Klicken Sie auf Deploy (Bereitstellen), um die Bereitstellungsrezepte auf den angegebenen Instances auszuführen, welche die Bereitstellungsseite anzeigt. Wenn der Vorgang abgeschlossen ist, markiert AWS OpsWorks Stacks die App mit einem grünen Häkchen, was auf eine erfolgreiche Bereitstellung hinweist. Wenn die Bereitstellung fehlschlägt, markiert AWS OpsWorks Stacks die App mit einem roten X. In diesem Fall können Sie auf der Seite Bereitstellungen das Bereitstellungsprotokoll nach weiteren Informationen durchsuchen.



> **Note**
>
> Wenn Sie eine Aktualisierung für eine JSP-Anwendung bereitstellen, wird Tomcat die Aktualisierung möglicherweise nicht erkennen und stattdessen die vorhandene Anwendungsversion ausführen. Dies kann beispielsweise auftreten, wenn Sie die Anwendung als ZIP-Datei bereitstellen, die nur eine JSP-Seite enthält. Um sicherzustellen, dass Tomcat die zuletzt bereitgestellte Version ausführt, muss das Stammverzeichnis des Projekts das Verzeichnis "WEB-INF" mit einer Datei `web.xml` enthalten. Der Inhalt einer `web.xml`-Datei kann vielfältig sein. Nachstehendes ist jedoch ausreichend, um sicherzustellen, dass Tomcat die Aktualisierungen erkennt und die aktuell bereitgestellte Anwendungsversion ausführt. Sie müssen die Version nicht für jede Aktualisierung ändern. Tomcat erkennt die Aktualisierung auch dann, wenn sich die Version nicht geändert hat.
>
> ```
> <context-param>
>   <param-name>appVersion</param-name>
>   <param-value>0.1</param-value>
> </context-param>
> ```

Andere Bereitstellungsbefehle

Die Seite Deploy app (App bereitstellen) enthält mehrere andere Befehle für die Verwaltung Ihrer Anwendungen und der damit verbundenen Server. Von den folgenden Befehlen ist nur Undeploy (Bereitstellung aufheben) für Anwendungen auf Chef 12-Stacks verfügbar.

Bereitstellung aufheben

Löst das Lebenszyklusereignis "Bereitstellung aufheben" aus, das die Rezepte zum Aufheben der Bereitstellung ausführt, um alle Versionen der Anwendung aus den angegebenen Instances zu entfernen.

Rollback

Stellt die zuvor bereitgestellte Anwendungsversion wieder her. Wenn Sie zum Beispiel die Anwendung dreimal bereitgestellt haben und dann Rollback ausführen, bietet der Server die Anwendung der zweiten Bereitstellung an. Wenn Sie Rollback erneut ausführen, bietet der Server die Anwendung der ersten Bereitstellung an. Standardmäßig speichert AWS OpsWorks Stacks die fünf letzten Bereitstellungen, sodass Sie bis zu vier Versionen rückgängig machen können. Wenn Sie die Anzahl der gespeicherten Versionen überschreiten, schlägt der Befehl fehl und stellt die älteste Version wieder her. Dieser Befehl ist in Chef 12-Stacks nicht verfügbar.

Starten des Webservers

Führt Rezepte aus, die den Anwendungsserver auf den angegebenen Instances starten. Dieser Befehl ist in Chef 12-Stacks nicht verfügbar.

Stoppen des Webservers

Führt Rezepte aus, die den Anwendungsserver auf den angegebenen Instances stoppen. Dieser Befehl ist in Chef 12-Stacks nicht verfügbar.

Neustarten des Webservers

Führt Rezepte aus, die den Anwendungsserver auf den angegebenen Instances neu starten. Dieser Befehl ist in Chef 12-Stacks nicht verfügbar.

Important

Start Web Server (Webserver starten), Stop Web Server (Webserver stoppen), Restart Web Server (Webserver neustarten) und Rollback sind im Wesentlichen benutzerdefinierte Versionen des Stack-Befehls "Execute Recipes". Sie führen eine Reihe von Rezepte aus, die die Aufgabe auf den angegebenen Instanzen ausführen.

- Diese Befehle lösen kein Lebenszyklusereignis aus, sodass Sie sie nicht zum Ausführen von benutzerdefiniertem Code anhängen können.
- Diese Befehle funktionieren nur für die integrierten Anwendungsserver-Ebenen.

Diese Befehle haben insbesondere keine Auswirkung auf benutzerspezifische Layer, auch wenn sie einen Anwendungsserver unterstützen. Zum Starten, Beenden oder Neustarten von Servern auf einer benutzerdefinierten Ebene müssen Sie benutzerdefinierte Rezepte implementieren und den Stack-Befehl "Execute Recipes" verwenden, um diese auszuführen. Weitere Informationen zur Implementierung und Installation von benutzerspezifischen Rezepten finden Sie unter Cookbooks und Rezepte.

Bearbeiten von Anwendungen

Important

Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können eine Anwendungskonfiguration ändern, indem Sie die Anwendung bearbeiten. Wenn Sie beispielsweise bereit sind, eine neue Version bereitzustellen, können Sie die AWS OpsWorks Stacks-Einstellungen der App bearbeiten, um den neuen Repository-Zweig zu verwenden. Sie müssen über Verwaltungs- oder Bereitstellungsberechtigungen verfügen, wenn Sie eine Anwendungskonfiguration bearbeiten möchten. Weitere Informationen finden Sie unter Verwalten von Benutzerberechtigungen.

Bearbeiten einer Anwendung

1. Klicken Sie auf der Seite Apps auf den Namen der Anwendung, um die Detailseite zu öffnen.
2. Klicken Sie auf Edit (Bearbeiten), um die Anwendungskonfiguration zu ändern.

   - Wenn Sie den Namen der App ändern, verwendet AWS OpsWorks Stacks den neuen Namen, um die App in der Konsole zu identifizieren.

     Eine Änderung des Namens ändert nicht den dazugehörigen Kurznamen. Der Kurzname wird eingerichtet, wenn Sie die Anwendung dem Stack hinzufügen, und kann im Nachhinein nicht mehr geändert werden.

- Wenn Sie eine geschützte Umgebungsvariable angegeben haben, können Sie den Wert weder anzeigen noch bearbeiten. Sie können jedoch einen neuen Wert angeben, indem Sie auf Update value (Wert aktualisieren) klicken.

3. Klicken Sie auf Save (Speichern), um die neue Konfiguration zu speichern, und dann auf Deploy App (App bereitstellen), um die Anwendung bereitzustellen.

Das Bearbeiten einer App ändert die Einstellungen von AWS OpsWorks Stacks, hat jedoch keine Auswirkungen auf die Instanzen des Stacks. Wenn Sie zum ersten Mal eine Anwendung bereitstellen, laden die Bereitstellungsrezepte den Code und die zugehörigen Dateien auf die Instances des Anwendungsservers herunter, die dann eine lokale Kopie ausführen. Wenn Sie die App im Repository ändern oder andere Einstellungen ändern, müssen Sie die App bereitstellen, um die Updates auf Ihren App-Serverinstanzen wie folgt zu installieren. AWS OpsWorks Stacks stellt die aktuelle App-Version automatisch auf neuen Instanzen bereit, wenn diese gestartet werden. Für vorhandene Instances ist die Situation jedoch eine andere:

- AWS OpsWorks Stacks stellt die aktuelle App-Version automatisch auf neuen Instanzen bereit, wenn diese gestartet werden.
- AWS OpsWorks Stacks stellt automatisch die neueste App-Version für Offline-Instanzen bereit, einschließlich lastbasierter und zeitbasierter Instanzen, wenn diese neu gestartet werden.
- Für Online-Instances müssen Sie die aktualisierte Anwendung manuell bereitzustellen.

Weitere Informationen zum Bereitstellen von Anwendungen finden Sie unter Bereitstellen von Anwendungen.

Verbinden einer Anwendung mit einem Datenbankserver

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können einen Amazon RDS-Datenbankserver einer App zuordnen, wenn Sie die App erstellen oder später, indem Sie die App bearbeiten. Ihre Anwendung kann dann die Datenbankverbindungsinformationen verwenden — Benutzername, Passwort,... — um eine Verbindung zum Datenbankserver herzustellen. Wenn Sie eine App bereitstellen, stellt AWS OpsWorks Stacks diese Informationen Anwendungen auf zwei Arten zur Verfügung:

- Für Linux-Stacks erstellt AWS OpsWorks Stacks auf jeder integrierten Anwendungsserver-Instance eine Datei mit den Verbindungsdaten zum Datenbankserver, auf die die Anwendung zugreifen kann.
- AWS OpsWorks Stacks enthält die Verbindungsinformationen in der Stack-Konfiguration und in den Bereitstellungsattributen, die auf jeder Instanz installiert sind.

  Sie können ein benutzerdefiniertes Rezept implementieren, um die Verbindungsinformationen aus diesen Attributen zu extrahieren und in einer Datei in Ihrem bevorzugten Format zu speichern. Weitere Informationen finden Sie unter Übermitteln von Daten an Anwendungen.

> **Important**
>
> Wenn Sie für Linux-Stacks einen Amazon RDS-Service-Layer mit Ihrer App verknüpfen möchten, müssen Sie das entsprechende Treiberpaket wie folgt zur zugehörigen App-Serverschicht hinzufügen:
>
> 1. Klicken Sie im Navigationsbereich auf Layers (Ebenen) und öffnen Sie die Registerkarte Recipes (Rezepte) des Anwendungsservers.
> 2. Klicken Sie auf Edit (Bearbeiten) und fügen Sie das entsprechende Treiberpaket zu OS Packages (OS-Pakete) hinzu. Legen Sie beispielsweise `mysql` fest, wenn der Layer Amazon Linux-Instances enthält, und `mysql-client`, wenn der Layer Ubuntu-Instances enthält.
> 3. Speichern Sie die Änderungen und stellen Sie die Anwendung erneut bereit.

Verwenden von benutzerdefinierten Rezepten

Sie können ein benutzerdefiniertes Rezept implementieren, um die Verbindungsdaten aus den `deploy`-Attributen der App zu extrahieren und in einer für die Anwendung lesbaren Form, beispielsweise in einer YAML-Datei, zu speichern.

Sie können einer App entweder beim Erstellen der App oder jederzeit später durch Bearbeiten der App einen Datenbankserver zuweisen. Wenn Sie die App bereitstellen, installiert AWS OpsWorks Stacks auf jeder Instance eine Stack-Konfiguration und Bereitstellungsattribute, die die Datenbankverbindungsinformationen enthalten. Ihre App kann diese Attribute dann abrufen. Im Detail hängt dieser Vorgang davon ab, ob Sie einen Linux- oder Windows-Stack verwenden.

Verbinden des Datenbankservers für einen Linux-Stack

Bei Linux-Stacks umfasst der `deploy` Namespace der Stack-Konfiguration und der Bereitstellungsattribute ein Attribut für jede bereitgestellte App, das mit dem Kurznamen der App benannt ist. Wenn Sie einen Datenbankserver an eine App anhängen, füllt AWS OpsWorks Stacks das `[:database]` App-Attribut mit den Verbindungsinformationen und installiert es für jede nachfolgende Bereitstellung auf den Instanzen des Stacks. Die Attributwerte werden entweder vom Benutzer bereitgestellt oder von AWS OpsWorks Stacks generiert.

> **Note**
>
> AWS OpsWorks Stacks ermöglicht es Ihnen, einen Datenbankserver an mehrere Apps anzuhängen, aber jede App kann nur einen angeschlossenen Datenbankserver haben. Wenn Sie eine Anwendung mit mehreren Datenbankservern verbinden möchten, ordnen Sie einen der Server der App zu und verbinden Sie die App anhand der `deploy`-Attribute mit diesem Server. Übergeben Sie mithilfe von JSON die Verbindungsinformationen für die anderen Datenbankserver an die Anwendung. Weitere Informationen finden Sie unter Übermitteln von Daten an Anwendungen.

Eine Anwendung kann die Verbindungsinformationen aus den `deploy`-Attributen der Instance nutzen, um sich mit einer Datenbank zu verbinden. Anwendungen können jedoch nicht direkt auf diese Informationen zugreifen — nur Rezepte können auf die Attribute zugreifen. `deploy` Implementieren Sie dafür ein benutzerdefiniertes Rezept, das die Verbindungsinformationen aus den `deploy`-Attributen extrahiert und sie in einer Datei speichert, die von der Anwendung gelesen werden kann.

Verwenden von -Umgebungsvariablen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir

empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Die Empfehlungen in diesem Abschnitt gelten für Chef 11.10 und frühere Versionen. Wenn Sie in Chef-12 und höheren Releases Umgebungsvariablen abrufen möchten, müssen Sie den Data Bag der Anwendung verwenden. Weitere Informationen finden Sie unter AWS OpsWorks Data Bag Reference und App Data Bag (aws_opsworks_app).

Wenn Sie Umgebungsvariablen für eine App angeben, fügt AWS OpsWorks Stacks die Variablendefinitionen zu den Attributen der App hinzu. `deploy`

Benutzerdefinierte Layers können mit einem Rezept den Variablenwert abrufen, indem sie die Standard-Knotensyntax verwenden und in einem Formular speichern, auf das die Layer-Anwendungen zugreifen können.

Sie müssen ein benutzerdefiniertes Rezept implementieren, das die Umgebungsvariable aus den `deploy` Attributen der Instance abruft. Das Rezept kann dann die Daten auf der Instance in einem Format speichern, auf das die Anwendung zugreifen kann (z. B. eine YAML-Datei). Die Definitionen der Umgebungsvariablen einer Anwendung werden in den `deploy` Attributen in den `environment_variables` der Anwendung gespeichert. Das folgende Beispiel zeigt den Speicherort dieser Attribute für eine Anwendung mit dem Namen `simplephpapp`, wobei JSON zur Darstellung der Attributstruktur verwendet wird.

```
{
  ...
  "ssh_users": {
  },
  "deploy": {
    "simplephpapp": {
      "application": "simplephpapp",
      "application_type": "php",
      "environment_variables": {
        "USER_ID": "168424",
        "USER_KEY": "somepassword"
```

```
      },
    ...
  }
}
```

Ein Rezept kann Variablenwerte über die Standard-Knotensyntax abrufen. Das folgende Beispiel zeigt, wie Sie den Wert `USER_ID` aus dem vorangegangenen JSON-Format abrufen und im Chef-Protokoll speichern.

```
Chef::Log.info("USER_ID: #{node[:deploy]['simplephpapp'][:environment_variables]
[:USER_ID]}")
```

Eine detaillierte Beschreibung, wie Sie Informationen aus der Stack-Konfiguration abrufen, JSON-Informationen bereitstellen und auf der Instance speichern, finden Sie unter Übermitteln von Daten an Anwendungen.

Übermitteln von Daten an Anwendungen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Es ist häufig nützlich, Daten wie z. B. Schlüssel-Wert-Paare an eine Anwendung auf dem Server zu übermitteln. Verwenden Sie dazu das benutzerdefinierte JSON-Objekt, um Daten zu einem Stack hinzuzufügen. AWS OpsWorks Stacks fügt die Daten für jedes Lebenszyklusereignis dem Knotenobjekt jeder Instanz hinzu.

Beachten Sie jedoch, dass die Anwendungen im Gegensatz zu den Rezepten nicht in der Lage sind, die benutzerdefinierten JSON-Daten mithilfe von Chef-Attributen abzurufen. Eine Möglichkeit zur Übermittlung von benutzerdefinierten JSON-Daten an eine oder mehrere Anwendungen ist die Implementierung eines benutzerdefinierten Rezepts, die die Daten aus dem node-Objekt extrahiert und in eine von der Anwendung lesbare Datei schreibt. Mit dem in diesem Thema aufgeführten Beispiel wird erläutert, wie Daten in eine YAML-Datei geschrieben werden. Sie können denselben Grundansatz für andere Formate wie JSON oder XML verwenden.

Um Schlüssel-Wert-Daten an die Stack-Instances zu übermitteln, fügen Sie ein benutzerdefiniertes JSON-Objekt folgendermaßen zum Stack hinzu. Weitere Informationen zum Hinzufügen eines benutzerdefinierten JSON-Objekts zu einem Stack finden Sie unter Nutzen eines benutzerdefinierten JSON-Objekts.

```
{
  "my_app_data": {
    "app1": {
      "key1": "value1",
      "key2": "value2",
      "key3": "value3"
    },
    "app2": {
      "key1": "value1",
      "key2": "value2",
      "key3": "value3"
    }
  }
}
```

In diesem Beispiel wird angenommen, dass Sie über zwei Anwendungen mit den Kurznamen `app1` und `app2` mit jeweils drei Datenwerten verfügen. Das zugehörige Rezept geht davon aus, dass Sie die zugehörigen Daten mithilfe des Kurznamens der Anwendung identifizieren; die restlichen Namen sind willkürlich. Weitere Informationen zu Kurznamen für Anwendungen finden Sie unter Einstellungen.

Das Rezept im folgenden Beispiel zeigt, wie die Daten für die einzelnen Anwendungen aus den `deploy`-Attributen extrahiert und in eine `.yml`-Datei gespeichert werden. Das Rezept geht davon aus, dass Ihr benutzerdefiniertes JSON-Objekt für jede Anwendung Daten enthält.

```
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
```

Die `deploy`-Attribute enthalten ein Attribut für jede Anwendung mit deren Kurznamen. Jedes Anwendungsattribut enthält eine Reihe von Attributen mit einer Vielzahl von Informationen über

die Anwendung. Dieses Beispiel verwendet das Bereitstellungsverzeichnis der Anwendung, das anhand des `[:deploy][:app_short_name][:deploy_to]`-Attributs definiert wird. Weitere Informationen zu `[:deploy]` finden Sie unter Bereitstellungsattribute.

Für jede Anwendung in `deploy` führt das Rezept Folgendes aus:

1. Eine Datei namens `app_data.yml` wird im Unterverzeichnis `shared/config` des Verzeichnisses `[:deploy_to]` der Anwendung erstellt.

   Weitere Informationen darüber, wie AWS OpsWorks Stacks Apps installiert, finden Sie unter. Bereitstellungsrezepte
2. Sie konvertiert die benutzerdefinierten JSON-Daten der Anwendung zu YAML und schreibt die formatierten Daten in `app_data.yml`.

So übermitteln Sie Daten an eine Anwendung

1. Fügen Sie eine Anwendung zum Stack hinzu und notieren Sie deren Kurznamen. Weitere Informationen finden Sie unter Hinzufügen von Apps.
2. Fügen Sie die benutzerdefinierten JSON-Werte zusammen mit den Anwendungsdaten zu den `deploy`-Attributen hinzu, wie zuvor beschrieben. Weitere Informationen zum Hinzufügen eines benutzerdefinierten JSON-Objekts zu einem Stack finden Sie unter Nutzen eines benutzerdefinierten JSON-Objekts.
3. Erstellen Sie ein Rezeptbuch und fügen Sie zu diesem ein Rezept mit dem auf dem vorgenannten Beispiel basierenden Code hinzu, der entsprechend den im benutzerdefinierten JSON-Objekt verwendeten Attributnamen modifiziert ist. Weitere Informationen zum Erstellen von Rezeptbüchern und Rezepten finden Sie unter Cookbooks und Rezepte. Wenn Sie bereits über benutzerdefinierte Rezeptbücher für diesen Stack verfügen, können Sie das Rezept auch einem vorhandenen Rezeptbuch oder den Code sogar einem vorhandenen Bereitstellungsrezept hinzufügen.
4. Installieren Sie das Rezeptbuch auf Ihrem Stack. Weitere Informationen finden Sie unter Installieren von benutzerdefinierten Rezeptbüchern.
5. Weisen Sie das Rezept dem Deploy-Lifecycle-Ereignis des App-Server-Layers zu. AWS OpsWorks Stacks führt das Rezept dann auf jeder neuen Instanz aus, nachdem diese gestartet wurde. Weitere Informationen finden Sie unter Ausführen von Rezepten.
6. Stellen Sie die Anwendung bereit, die auch Stack-Konfigurations- und Bereitstellungsattribute mit Ihren Daten installiert.

Note

Wenn die Datendateien zur Verfügung stehen müssen, bevor die Anwendung bereitgestellt wird, können Sie das Rezept auch dem Einrichtungs-Lebenszyklusereignis des Layers zuweisen, welcher unmittelbar nach Beendigung des Bootvorgangs eintritt. AWS OpsWorks Stacks hat die Bereitstellungsverzeichnisse jedoch noch nicht erstellt, daher sollte Ihr Rezept die erforderlichen Verzeichnisse explizit erstellen, bevor Sie die Datendatei erstellen. Im folgenden Beispiel wird explizit das Verzeichnis `/shared/config` der Anwendung und anschließend die Datendatei in diesem Verzeichnis erstellt.

```
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
```

Zum Laden der Daten können Sie beispielsweise folgenden Sinatra-Code verwenden:

```
#!/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
```

Sie können die Anwendungsdaten jederzeit durch die Aktualisierung des benutzerdefinierten JSON-Objekts wie folgt aktualisieren.

So aktualisieren Sie die Anwendungsdaten

1. Bearbeiten Sie das benutzerdefinierte JSON-Objekt, um die Daten zu aktualisieren.
2. Stellen Sie die App erneut bereit, wodurch AWS OpsWorks Stacks angewiesen wird, die Deploy-Rezepte auf den Instanzen des Stacks auszuführen. Die Rezepte greifen auf aktualisierte Stack-Konfigurations- und Bereitstellungsattribute zurück, sodass Ihr benutzerdefiniertes Rezept die Datendateien mit den gegenwärtigen Werten aktualisiert.

Verwenden von Git-Repository-SSH-Schlüsseln

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Beim Git-Repository-SSH-Schlüssel, der manchmal auch als SSH-Bereitstellungsschlüssel bezeichnet wird, handelt es sich um einen SSH-Schlüssel ohne Passwort, der Zugriff auf ein privates Git-Repository ermöglicht. Im Idealfall gehört er nicht einem bestimmten Entwickler. Ziel ist es, AWS OpsWorks Stacks die asynchrone Bereitstellung von Apps oder Kochbüchern aus einem Git-Repository zu ermöglichen, ohne dass weitere Eingaben von Ihnen erforderlich sind.

Im Folgenden werden die grundlegenden Schritte zum Erstellen eines Repository-SSH-Schlüssels erläutert. Weitere Informationen finden Sie in der Dokumentation Ihres Repositorys. Zum Beispiel beschreibt Managing Deploy Keys, wie du einen Repository-SSH-Schlüssel für ein GitHub Repository erstellst, und Deployment Keys auf Bitbucket beschreibt, wie du einen Repository-SSH-Schlüssel für ein Bitbucket-Repository erstellst. Beachten Sie, dass in einigen Dokumentationen die Erstellung eines Schlüssels auf einem Server erläutert wird. Ersetze bei AWS OpsWorks Stacks in der Anleitung einfach „Server“ durch „Workstation“.

So erstellen Sie einen Repository-SSH-Schlüssel

1. Erstellen Sie mithilfe eines Programms wie `ssh-keygen` ein SSH-Bereitstellungsschlüsselpaar für Ihr Git-Repository auf Ihrer Workstation.

   > **Important**
   >
   > AWS OpsWorks Stacks unterstützt keine SSH-Schlüssel-Passphrasen.

2. Weisen Sie den öffentlichen Schlüssel dem Repository zu und speichern Sie den privaten Schlüssel auf Ihrer Workstation.
3. Geben Sie den privaten Schlüssel im Feld Repository SSH Key (Repository-SSH-Schlüssel) ein, wenn Sie eine Anwendung hinzufügen oder ein Rezeptbuch-Repository festlegen. Weitere Informationen finden Sie unter Hinzufügen von Apps.

AWS OpsWorks Stacks übergibt den SSH-Schlüssel des Repositorys an jede Instanz, und die integrierten Rezepte verwenden dann den Schlüssel, um eine Verbindung zum Repository herzustellen und den Code herunterzuladen. Der Schlüssel wird in den `deploy`-Attributen als `node[:deploy]['appshortname'][:scm][:ssh_key]` gespeichert und ist nur für den Root-Benutzer zugänglich.

Verwenden von benutzerdefinierten Domänen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn Sie einen Domänennamen bei einem Drittanbieter hosten, können Sie diesen Domänennamen einer App zuordnen. Gehen Sie dabei wie folgt vor:

1. Erstellen Sie eine Subdomain bei Ihrem DNS-Registrar und ordnen Sie sie der Elastic IP-Adresse Ihres Load Balancers oder der öffentlichen IP-Adresse Ihres Anwendungsservers zu.

2. Aktualisieren Sie Ihre App-Konfiguration, um auf die Subdomain zu verweisen, und stellen Sie die App erneut bereit.

> **Note**
>
> Leiten Sie den nicht qualifizierten Domänennamen (z. B. meineapp1.beispiel.de) an Ihren qualifizierten Domänennamen (z. B. www.meineapp1.beispiel.de) weiter, damit beide Ihrer App zugeordnet werden.

Wenn Sie eine Domäne für eine App konfigurieren, wird diese als Server-Alias in der Serverkonfigurationsdatei gespeichert. Wenn Sie einen Load Balancer verwenden, überprüft dieser den Domänennamen bei eingehenden Anfragen in der URL und leitet den Datenverkehr anhand der Domäne weiter.

So ordnen Sie eine Subdomain einer IP-Adresse zu

1. Wenn Sie einen Load Balancer verwenden, klicken Sie auf der Seite Instances auf die Load Balancer-Instance, um deren Detailseite zu öffnen und die Elastic IP-Adresse der Instance abzurufen. Sie können andernfalls auch die öffentliche IP-Adresse auf der Detailseite der Anwendungsserver-Instance abrufen.
2. Befolgen Sie die Anweisungen Ihres DNS-Registrars, um Ihre Subdomain zu erstellen und der IP-Adresse aus Schritt 1 zuzuordnen.

> **Note**
>
> Wenn die Load Balancer-Instance beendet wird, erhalten Sie eine neue Elastic IP-Adresse. Sie müssen dann die DNS-Registrar-Einstellungen aktualisieren, um auf diese neue Elastic IP-Adresse zu verweisen.

AWS OpsWorks Stacks fügt einfach die Domain-Einstellungen zu den Attributen der App hinzu. `deploy` Um die Informationen aus dem Knotenobjekt abzurufen und den Server zu konfigurieren, müssen Sie ein benutzerdefiniertes Rezept implementieren. Weitere Informationen finden Sie unter Cookbooks und Rezepte.

Ausführen von mehreren Anwendungen auf demselben Anwendungsserver

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Die Informationen in diesem Thema gelten nicht für Node.js-Apps.

Wenn Sie mehrere Anwendungen desselben Typs haben, kann es unter Umständen kostengünstiger sein, diese auf denselben Anwendungsserver-Instances auszuführen.

So führen Sie mehrere Anwendungen auf demselben Server aus

1. Fügen Sie dem Stack für jede Anwendung eine App hinzu.
2. Erstellen Sie für jede App eine eigene Subdomain und ordnen Sie die Subdomains der IP-Adresse des Anwendungsservers oder des Load Balancers zu.
3. Bearbeiten Sie die Konfiguration der einzelnen Apps und geben Sie dort die entsprechenden Subdomains ein.

Weitere Informationen zu diesen Aufgaben finden Sie unter Verwenden von benutzerdefinierten Domänen.

> **Note**
>
> Wenn auf Ihren Anwendungsservern mehrere HTTP-Anwendungen ausgeführt werden, können Sie Elastic Load Balancing für den Lastenausgleich verwenden. Bei mehreren HTTPS-Anwendungen müssen Sie entweder die SSL-Verbindung auf dem Load Balancer beenden oder für jede Anwendung einen eigenen Stack erstellen. HTTPS-Anfragen sind verschlüsselt. Wenn Sie daher die SSL-Verbindung auf den Servern beenden, kann der Load

Balancer den Domänennamen nicht überprüfen, um zu bestimmen, welche Anwendung die Anfrage bearbeiten soll.

Verwenden von SSL

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn Sie SSL in Ihrer Anwendung nutzen möchten, müssen Sie zunächst ein digitales Serverzertifikat von einer Zertifizierungsstelle einholen. Der Einfachheit halber wird in dieser Anleitung ein eigenes Zertifikat erstellt und signiert. Selbstsignierte Zertifikate sind für Lern- und Testzwecke hilfreich. Für Produktions-Stacks sollten jedoch grundsätzlich von einer Zertifizierungsstelle signierte Zertifikate verwendet werden.

In dieser Anleitung werden Sie Folgendes tun:

1. Installieren und konfigurieren Sie OpenSSL.
2. Erstellen eines privaten Schlüssels
3. Erstellen einer Zertifikatssignierungsanforderung
4. Generieren eines selbstsignierten Zertifikats
5. Bearbeiten der Anwendung mit Ihren Zertifikatsinformationen

Important

Wenn in Ihrer Anwendung SSL genutzt wird, sollten Sie in den Anwendungsserver-Ebenen nach Möglichkeit SSLv3 deaktivieren, um die Schwachstellen zu vermeiden, die unter CVE-2014-3566 beschrieben sind. Wenn Ihr Stack eine Ganglia-Schicht enthält, sollten Sie SSL v3 auch für diese Ebene deaktivieren. Im Einzelnen hängt dies von dem jeweiligen Layer ab, wie nachfolgend genauer beschrieben.

- AWS OpsWorks Stacks-Schicht für Java App Server
- Node.js App Server AWS OpsWorks Stacks Layer
- PHP-App-Server: AWS OpsWorks Stacks-Layer
- Rails App Server: AWS OpsWorks Stacks-Ebene
- Statischer Webserver: AWS OpsWorks Stacks Layer
- Ganglien-Schicht

Themen

Schritt 1: Installieren und Konfigurieren von OpenSSL

Um Serverzertifikate erstellen und hochladen zu können, muss Ihr Tool die Protokolle SSL und TLS unterstützen. Das Open-Source-Tool OpenSSL bietet die grundlegenden Verschlüsselungsfunktionen, die Sie zum Erstellen eines RSA-Tokens und Signieren des Tokens mit Ihrem privaten Schlüssel benötigen.

Im folgenden Verfahren gehen wir davon aus, dass auf Ihrem Computer noch kein OpenSSL installiert ist.

So installieren Sie OpenSSL unter Linux und Unix

1. Rufen Sie OpenSSL: Source, Tarballs (OpenSSL: Quelle, Tarballs) auf.
2. Laden Sie die aktuelle Quelldatei herunter.
3. Erstellen Sie das Paket.

So installieren Sie OpenSSL unter Windows

1. Wenn das Microsoft Visual C++ 2008 Redistributable Package noch nicht auf Ihrem System installiert ist, laden Sie das Paket herunter.

2. Führen Sie das Installationsprogramm aus und befolgen Sie die Anweisungen des Microsoft Visual C++ 2008 Redistributable-Installationsassistenten.
3. Rufen Sie OpenSSL: Binary Distributions (OpenSSL: Binäre Verteilungen) auf, klicken Sie auf die OpenSSL-Binärdatei für Ihre Umgebung und speichern Sie das Installationsprogramm lokal.
4. Führen Sie das Installationsprogramm aus und befolgen Sie die Anweisungen im OpenSSL Setup Wizard (OpenSSL-Einrichtungsassistent), um die Binärdateien zu installieren.

Erstellen Sie eine Umgebungsvariable, die auf den Installationspunkt von OpenSSL verweist. Öffnen Sie hierfür ein Terminal oder Befehlsfenster und geben Sie folgende Befehlszeilen ein.

- Unter Linux und Unix

```
export OpenSSL_HOME=path_to_your_OpenSSL_installation
```

- Unter Windows

```
set OpenSSL_HOME=path_to_your_OpenSSL_installation
```

Um den Pfad der OpenSSL-Binärdatei der Pfadvariablen Ihres Computers hinzuzufügen, öffnen Sie ein Terminal oder Befehlsfenster und geben Sie die folgenden Befehlszeilen ein.

- Unter Linux und Unix

```
export PATH=$PATH:$OpenSSL_HOME/bin
```

- Unter Windows

```
set Path=OpenSSL_HOME\bin;%Path%
```

> **Note**
>
> Alle Änderungen, die Sie mithilfe dieser Befehlszeilen an den Umgebungsvariablen vornehmen, sind nur für die aktuelle Befehlszeilensitzung gültig.

Schritt 2: Erstellen eines privaten Schlüssels

Sie benötigen einen eindeutigen privaten Schlüssel, um Ihre Zertifikatsignieranforderung zu erstellen. Sie können den Schlüssel mithilfe der folgenden Befehlszeile erstellen:

```
openssl genrsa 2048 > privatekey.pem
```

Schritt 3: Erstellen einer Zertifikatsignieranforderung

Ein Zertifikatsignieranforderung (Certificate Signing Request, CSR) ist eine Datei, die Sie an eine Zertifizierungsstelle senden, um ein digitales Serverzertifikat zu erhalten. Sie können die CSR-Anforderung mithilfe der folgenden Befehlszeile erstellen:

```
openssl req -new -key privatekey.pem -out csr.pem
```

Die Ausgabe für den Befehl sieht dann wie folgt aus:

```
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.
```

Mithilfe der folgenden Tabelle können Sie die Zertifikatsanforderung erstellen.

Daten für die Zertifikatsanforderung

Name	Beschreibung	Beispiel
Ländername	Die zweistellige ISO-Abkürzung für Ihr Land	US = USA
Bundesstaat oder Provinz	Der Name des Bundesstaats oder der Provinz, in dem bzw. der sich Ihre Organisation befindet. Dieser Name darf nicht abgekürzt werden.	Washington

Name	Beschreibung	Beispiel
Locality Name	Der Name der Stadt, in der sich Ihre Organisation befindet.	Seattle
Name der Organisation	Der vollständige, offizielle Name Ihrer Organisation. Kürzen Sie den Namen Ihrer Organisation nicht ab.	UnternehmenX
Organisationseinheit	(Optional) Für weitere Informationen zu Ihrer Organisation	Marketing
Common Name	Der vollqualifizierte Domänenname für Ihr CNAME. Bei der Überprüfung des Zertifikatsnamens wird eine Warnung angezeigt, wenn dieser nicht genau übereinstimmt.	www.example.com
E-Mail-Adresse	Die E-Mail-Adresse des Serveradministrators	someone@example.com

Note

Oft ist nicht klar, was der allgemeine Name ist, und dieses Feld wird dann falsch ausgefüllt. Der allgemeine Name ist in der Regel der Hostname zusammen mit dem Domänennamen. Er entspricht dem Schema "www.example.com" oder "example.com". CSR-Anfragen müssen den korrekten allgemeinen Namen enthalten.

Schritt 4: Einreichen der CSR-Anfrage bei der Zertifizierungsstelle

Für die Produktion müssen Sie eine CSR-Anfrage für ein Serverzertifikat bei einer Zertifizierungsstelle einreichen. Hierfür benötigen Sie möglicherweise weitere Informationen oder Identitätsnachweise. Wenn Ihr Antrag erfolgreich ist, erhalten Sie von der Zertifizierungsstelle ein digital signiertes Zertifikat und eventuell auch eine Zertifikatskettendatei. AWS empfiehlt keine bestimmte Zertifizierungsstelle. Eine unvollständige Liste der verfügbaren Zertifizierungsstellen finden Sie unter Certificate Authority - Providers auf Wikipedia.

Sie können auch ein selbstsigniertes Zertifikat für Testzwecke erstellen. Verwenden Sie für dieses Beispiel die folgende Befehlszeile, um ein selbstsigniertes Zertifikat zu erstellen.

```
openssl x509 -req -days 365 -in csr.pem -signkey privatekey.pem -out server.crt
```

Die Ausgabe sieht etwa folgendermaßen aus:

```
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
```

Schritt 5: Bearbeiten der App

Nachdem Sie Ihr Zertifikat generiert und signiert haben, müssen Sie in Ihrer App SSL aktivieren und die Zertifikatsinformationen eintragen. Wählen Sie auf der Seite Apps eine App aus, um die Detailseite aufzurufen, und klicken Sie dann auf Edit App (App bearbeiten). Wählen Sie für Enable SSL (SSL aktivieren) die Option Yes (Ja) aus, um SSL zu aktivieren. Es werden folgende Konfigurationsoptionen angezeigt.

SSL Certificate (SSL-Zertifikat)

Kopieren Sie den Inhalt der CRT-Zertifikatsdatei für den öffentlichen Schlüssel in das Feld. Das Zertifikat sollte etwa wie folgt aussehen:

```
-----BEGIN CERTIFICATE-----
MIICuTCCAiICCQCtqFKItVQJpzANBgkqhkiG9w0BAQUFADCBoDELMAkGA1UEBhMC
dXMxEzARBgNVBAgMCndhc2hpbmd0b24xEDAOBgNVBAcMB3NlYXR0bGUxDzANBgNV
BAoMBmFtYXpvbjEWMBQGA1UECwwNRGV2IGFuZCBUb29sczEdMBsGA1UEAwwUc3Rl
cGhhbmllYXBpZXJjZS5jb20xIjAgBgkqhkiG9w0BCQEWE3NhcGllcmNlQGFtYXpv
```

```
...
-----END CERTIFICATE-----
```

> **Note**
>
> Wenn Sie Nginx verwenden und eine Zertifikatskettendatei haben, hängen Sie deren Inhalt an die Zertifikatsdatei für den öffentlichen Schlüssel an.

Wenn Sie ein vorhandenes Zertifikat aktualisieren, gehen Sie wie folgt vor:

- Wählen Sie Update SSL certificate (SSL-Zertifikat aktualisieren) aus, um das Zertifikat zu aktualisieren.
- Falls das neue Zertifikat nicht mit dem vorhandenen privaten Schlüssel übereinstimmt, wählen Sie Update SSL certificate key (SSL-Zertifikatschlüssel aktualisieren) aus.
- Wenn das neue Zertifikat nicht mit der vorhandenen Zertifikatskette übereinstimmt, wählen Sie Update SSL certificates (SSL-Zertifikate aktualisieren) aus.

SSL Certificate Key (SSL-Zertifikatschlüssel)

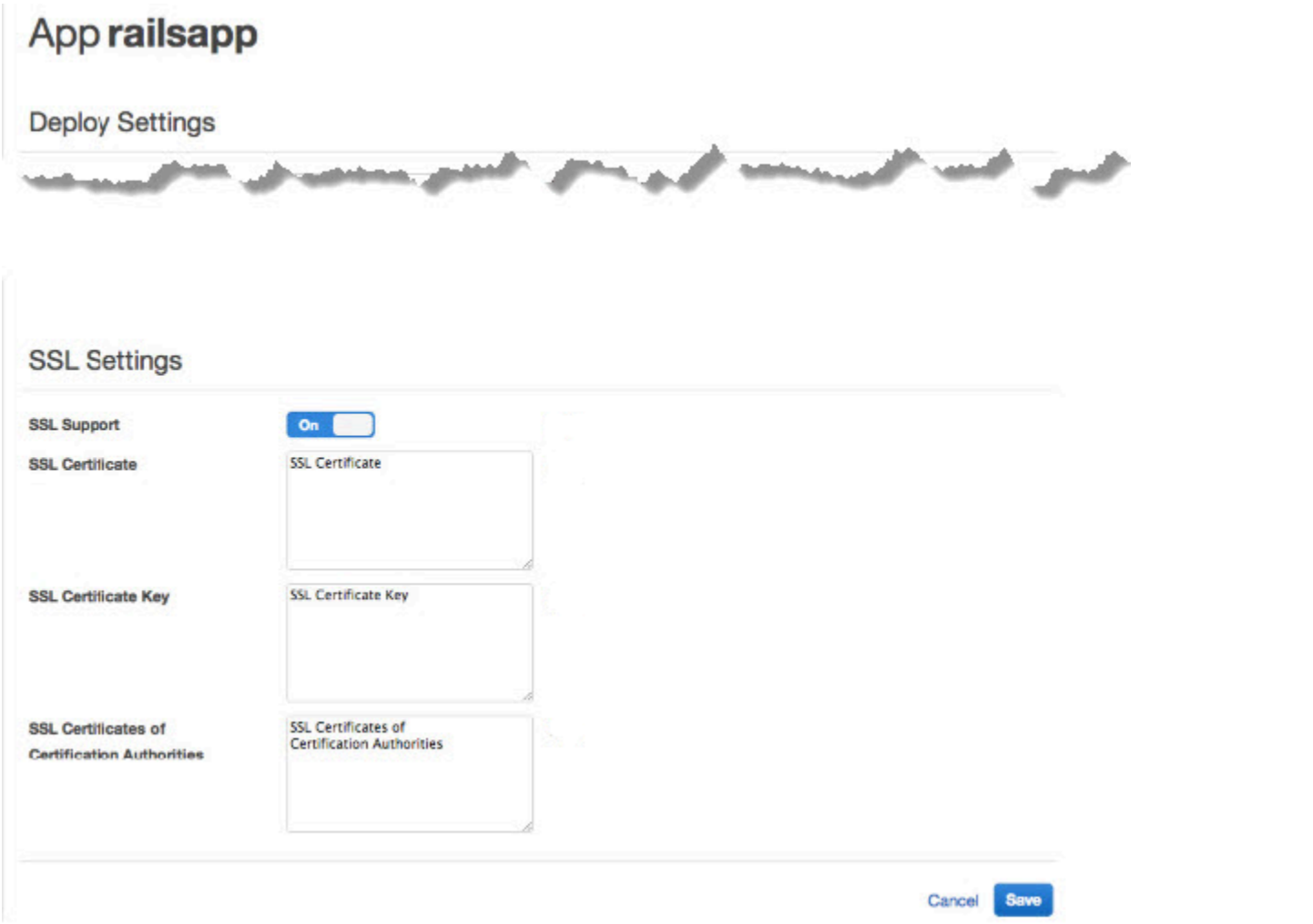
Kopieren Sie den Inhalt der privaten Schlüsseldatei (PEM-Datei) in das Feld. Es sollte etwa wie folgt aussehen:

```
----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQC0CYklJY5r4vV2NHQYEpwtsLuMMBhylMrgBShKq+HHVLYQQCL6
+wGIiRq5qXqZlRXje3GM5Jvcm6q0R71MfRIl1FuzKyqDtneZaAIEYniZibHiUnmO
/UNqpFDosw/6hY3ONk0fSBlU4ivD0Gjpf6J80jL3DJ4R23Ed0sdL4pRT3QIDAQAB
AoGBAKmMfWrNRqYVtGKgnWB6Tji9QrKQLMXjmHeGg95mppdJELiXHhpMvrHtpIyK
...
-----END RSA PRIVATE KEY-----
```

SSL certificates of Certification Authorities (SSL-Zertifikate von Zertifizierungsstellen)

Falls Sie eine Zertifikatskettendatei haben, kopieren Sie deren Inhalt in das Feld.

> **Note**
>
> Falls Sie Nginx verwenden, muss dieses Feld leer bleiben. Wenn Sie eine Zertifikatskettendatei haben, hängen Sie diese über SSL Certificate (SSL-Zertifikat) an die Zertifikatsdatei des öffentlichen Schlüssels an.



Klicken Sie auf Save (Speichern) und stellen Sie die App erneut bereit, um die Online-Instances zu aktualisieren.

Für die integrierten Anwendungsserverschichten aktualisiert AWS OpsWorks Stacks automatisch die Serverkonfiguration. Nach der Bereitstellung können Sie die Installation von OpenSSL wie nachfolgend beschrieben überprüfen.

So überprüfen Sie die OpenSSL-Installation

1. Rufen Sie die Seite Instances auf.
2. Klicken Sie auf die IP-Adresse der Anwendungsserver-Instance oder die IP-Adresse des Load Balancers (bei Verwendung eines Load Balancers), um die App auszuführen.
3. Ändern Sie das Präfix der IP-Adresse von **`http://`** in **`https://`** und aktualisieren Sie den Browser, um zu überprüfen, ob die Seite mit SSL korrekt geladen wird.

Benutzer, die Apps für die Ausführung in Mozilla Firefox konfiguriert haben, erhalten gelegentlich den folgenden Zertifikatsfehler: `SEC_ERROR_UNKNOWN_ISSUER`. Dieser Fehler kann durch Zertifikatersetzung in den Antivirus- und Anti-Malware-Programmen Ihrer Organisation, durch bestimmte Software zur Überwachung und Filterung des Netzwerkdatenverkehrs und durch Malware verursacht werden. Weitere Informationen zur Behebung dieses Fehlers finden Sie unter Beheben der Fehlercodes in der Meldung „Diese Verbindung ist nicht sicher“ auf sicheren Websites auf der Mozilla Firefox Support-Website.

Bei allen anderen Ebenen, einschließlich benutzerdefinierten Ebenen, fügt AWS OpsWorks Stacks die SSL-Einstellungen automatisch den `deploy`-Attributen der App hinzu. Um die Informationen aus dem Knotenobjekt abzurufen und den Server zu konfigurieren, müssen Sie ein benutzerdefiniertes Rezept implementieren.

Cookbooks und Rezepte

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks verwendet Chef-Kochbücher, um Aufgaben wie die Installation und Konfiguration von Paketen und die Bereitstellung von Apps zu erledigen. In diesem Abschnitt wird beschrieben, wie Kochbücher mit Stacks verwendet werden. AWS OpsWorks Weitere Informationen finden Sie unter Chef.

> **Note**
>
> AWS OpsWorks Stacks unterstützt derzeit die Chef-Versionen 12, 11.10.4, 11.4.4 und 0.9.15.5. Chef 0.9.15.5 ist jedoch veraltet und sollte für neue Stacks daher nicht mehr verwendet werden. Der Einfachheit halber wird auf sie nur mit den Haupt- und Nebenversionsnummern verwiesen. Stacks, auf denen Chef 0.9 oder 11.4 ausgeführt wird, verwenden Chef Solo. Auf Stacks mit Chef 12 oder 11.10 wird Chef Client im lokalen Modus ausgeführt. Für Linux-Stacks können Sie mit dem Configuration Manager angeben,

welche Chef-Version verwendet werden soll, wenn Sie einen Stack erstellen. Windows Stacks müssen Chef 12.2 verwenden. Weitere Informationen, einschließlich Richtlinien zum Migrieren von Stacks auf aktuelle Chef-Versionen, finden Sie unter Chef-Versionen.

Themen

Rezeptbuch-Repositorys

Important

Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Ihre benutzerdefinierten Rezeptbücher müssen in einem Online-Repository gespeichert sein, entweder in einem Archiv, wie z. B. einer ZIP-Datei, oder in einem Source Control Manager wie Git. Ein Stack kann nur ein benutzerdefiniertes Rezeptbuch-Repository haben. Das Repository kann jedoch eine beliebige Anzahl von Rezeptbüchern enthalten. Wenn Sie die Kochbücher installieren oder aktualisieren, installiert AWS OpsWorks Stacks das gesamte Repository in einem lokalen Cache auf jeder der Instanzen des Stacks. Wenn eine Instance z. B. ein oder mehrere Rezepte ausführen muss, verwendet sie den Code aus dem lokalen Cache.

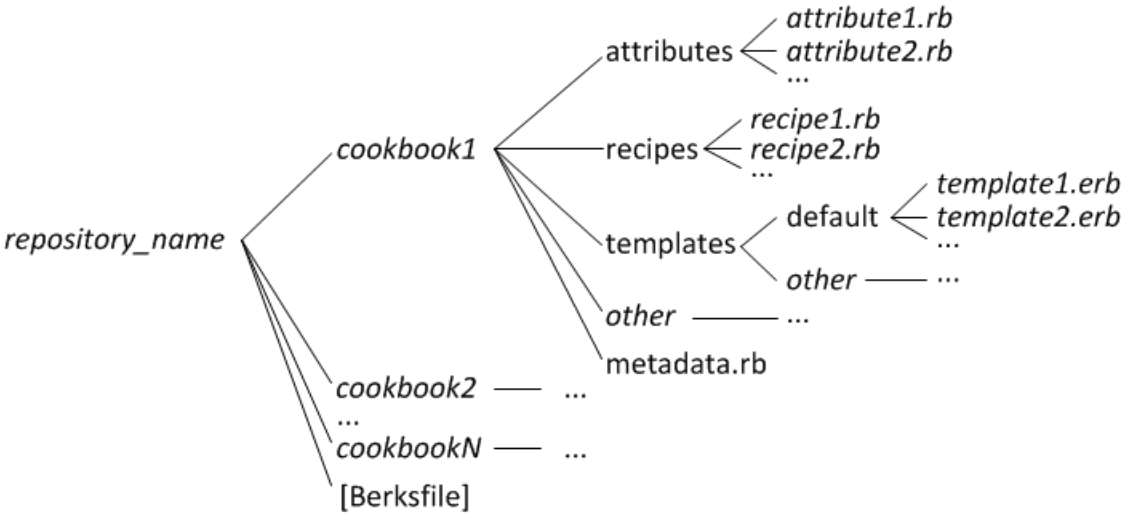
Im Folgenden wird beschrieben, wie Sie Ihr Rezeptbuch-Repository strukturieren, was abhängig vom Typ ist. Der kursive Text in den Abbildungen stellt benutzerdefinierte Verzeichnis- und Dateinamen dar, darunter der Repository- und Archivname.

Source Control Manager

AWS OpsWorks Stacks unterstützt die folgenden Quellcodeverwaltungsmanager:

- Linux-Stacks — Git und Subversion
- Windows-Stapel — Git

Im Folgenden sehen Sie das erforderliche Verzeichnis und die Dateistruktur:



- Die Rezeptbuch-Verzeichnisse müssen sich alle auf der obersten Ebene befinden.

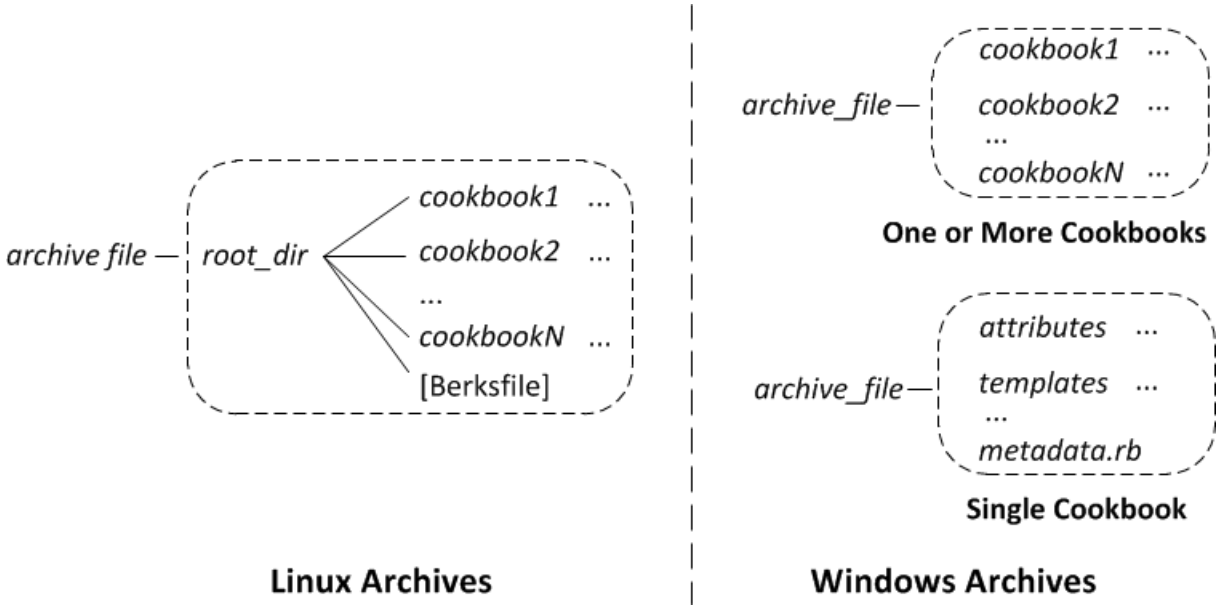
Archiv

AWS OpsWorks Stacks unterstützt die folgenden Archive:

- Linux-Stacks — Zip-, Gzip-, Bzip2- oder Tarball-Dateien, gespeichert auf Amazon S3 oder einer Website (HTTP-Archiv).

  AWS OpsWorks Stacks unterstützt keine unkomprimierten Tarballs.
- Windows-Stacks — Zip- und TGZ-Dateien (GZIP-komprimiertes Tar), gespeichert auf Amazon S3.

Im Folgenden wird das erforderliche Verzeichnis und die Dateistruktur angezeigt, was davon abhängt, ob Sie einen Linux- oder Windows-Stack ausführen. Die Rezeptbuch-Struktur ist die gleiche wie für SCM-Repositorys, sie wird also durch Auslassungszeichen dargestellt (...).



- Linux-Stacks — Die Kochbuchverzeichnisse müssen in einem Stammverzeichnis enthalten sein.
- Windows-Stacks — Die Kochbücher müssen sich auf der obersten Ebene des Archivs befinden.

  Wenn Sie nur ein Rezeptbuch haben, können Sie optional das Rezeptbuch-Verzeichnis weglassen und die Rezeptbuch-Dateien auf der obersten Ebene platzieren. In diesem Fall erhält AWS OpsWorks Stacks den Namen des Rezeptbuchs von metadata.rb.

Jedes Rezeptbuch-Verzeichnis verfügt über mindestens ein und in der Regel über alle der folgenden Standardverzeichnisse und Dateien, die Standardnamen verwenden müssen:

- `attributes`— Die Attributdateien des Kochbuches.
- `recipes`— Die Rezeptdateien des Kochbuches.
- `templates`— Die Vorlagendateien des Kochbuches.
- *`andere`* — Optionale benutzerdefinierte Verzeichnisse, die andere Dateitypen wie Definitionen oder Spezifikationen enthalten.
- `metadata.rb`— Die Metadaten des Kochbuches.

  Für Chef 11.10 und höher: Wenn Ihre Rezepte von anderen Rezeptbüchern abhängen, müssen Sie entsprechende `depends` Anweisungen in die Rezeptbuch-Datei `metadata.rb` einfügen. Wenn Ihr Rezeptbuch beispielsweise ein Rezept mit einem Statement wie `include_recipe anothercookbook::somerecipe` enthält, muss Ihre Rezeptbuch-Datei `metadata.rb` die

folgende Zeile enthalten: `depends "anothercookbook"`. Weitere Informationen finden Sie unter About Cookbook Metadata.

Vorlagen müssen in einem Unterverzeichnis des `templates`-Verzeichnisses gespeichert sein. Dieses enthält mindestens eine und optional mehrere Unterverzeichnisse. Diese Unterverzeichnisse können optional auch Unterverzeichnisse haben.

- Vorlagen haben in der Regel ein `default`-Unterverzeichnis. Dieses enthält die Vorlagendateien, die Chef standardmäßig benutzt.
- andere repräsentiert optionale Unterverzeichnisse, die für betriebssystemspezifische Vorlagen benutzt werden können.
- Chef verwendet automatisch die Vorlage aus dem entsprechenden Unterverzeichnis, basierend auf Benennungskonventionen, welche in File Specificity beschrieben werden. Für die Linux- und Ubuntu-Betriebssysteme können Sie beispielsweise betriebssystemspezifische Vorlagen in Unterverzeichnissen mit dem Namen `amazon`amazon oder `ubuntu` hinzufügen.

Die Details, wie Sie mit benutzerdefinierten Rezeptbüchern umgehen, hängen von Ihrem bevorzugten Repository-Typ ab.

So benutzen Sie ein S3-Archiv

1. Implementieren Sie Ihre Rezeptbücher mithilfe der im vorigen Abschnitt gezeigten Ordnerstruktur.
2. Erstellen Sie ein komprimiertes Archiv und laden Sie es in einen Amazon S3 S3-Bucket oder eine Website hoch.

   Wenn Sie Ihre Rezeptbücher aktualisieren, müssen Sie eine neue Archivdatei erstellen und hochladen. Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

So verwenden Sie eine SCM

1. Richten Sie ein Git- oder Subversion-Repository ein, indem Sie die bereits gezeigte Struktur verwenden.

2. Optional können Sie die Kontrollfunktionen der Repository-Version benutzen, um mehrere Branches oder Versionen zu implementieren.

   Wenn Sie Ihre Kochbücher aktualisieren, können Sie dies in einer neuen Filiale tun und einfach direkt OpsWorks die neue Version verwenden. Sie können auch bestimmte getaggte Versionen angeben. Details hierzu finden Sie unter Festlegen eines benutzerdefinierten Rezeptbuch-Repositorys.

Installieren von benutzerdefinierten Rezeptbüchernbeschreibt, wie AWS OpsWorks Stacks dein Kochbuch-Repository auf den Instanzen des Stacks installieren lässt.

> **Important**
>
> Nachdem Sie die vorhandenen Kochbücher im Repository aktualisiert haben, müssen Sie den Befehl `update_cookbooks` stack ausführen, um AWS OpsWorks Stacks anzuweisen, den lokalen Cache jeder Online-Instanz zu aktualisieren. Weitere Informationen finden Sie unter Ausführen von Stack-Befehlen.

Chef-Versionen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks unterstützt mehrere Versionen von Chef. Sie wählen die Version aus, wenn Sie den Stack erstellen. AWS OpsWorks Stacks installiert dann diese Version von Chef auf allen Instanzen des Stacks zusammen mit einer Reihe von integrierten Rezepten, die mit dieser Version kompatibel sind. Wenn Sie benutzerdefinierte Rezepte installieren, müssen diese mit der Chef-Version des Stacks kompatibel sein.

AWS OpsWorks Stacks unterstützt derzeit die Chef-Versionen 12, 11.10, 11.4 und 0.9 für Linux-Stacks und Chef 12.2 (derzeit Chef 12.22) für Windows-Stacks. Der Einfachheit halber wird auf sie

nur mit den Haupt- und Nebenversionsnummern verwiesen. Für Linux-Stacks können Sie mit dem Configuration Manager angeben, welche Chef-Version verwendet werden soll, wenn Sie einen Stack erstellen. Windows Stacks müssen Chef 12.2 verwenden. Weitere Informationen, einschließlich Richtlinien zum Migrieren von Stacks auf aktuelle Chef-Versionen, finden Sie unter Chef-Versionen. Vollständige Versionsinformationen finden Sie unter AWS OpsWorks Stacks-Betriebssysteme.

Chef 12.2

Die Unterstützung von Chef 12.2 wurde im Mai 2015 eingeführt und wird nur von Windows-Stacks verwendet. Die aktuelle Version von Chef auf Windows-Stacks ist Chef 12.22. Der Support läuft mit Ruby 2.3.6 und verwendet den Chef-Client im lokalen Modus. Dieser startet einen lokalen In-Memory-Chef-Server mit dem Namen chef-zero. Dieser Server ermöglicht Rezepten die Nutzung der Chef-Suchfunktion und Data Bags. Der Support hat einige Einschränkungen, die in Rezepte implementieren: Chef 12.2 beschrieben sind. Sie können jedoch viele Community-Rezeptbücher ohne Änderung ausführen.

Chef 12

Der Chef 12-Support wurde im Dezember 2015 eingeführt und wird nur für Linux-Stacks verwendet. Er läuft mit Ruby 2.1.6 oder 2.2.3 und verwendet den Chef-Client im lokalen Modus. Rezepte können so die Chef-Suchfunktion und Data Bags verwenden. Weitere Informationen finden Sie unter AWS OpsWorks Stacks-Betriebssysteme.

Chef 11.10

Der Chef 11.10-Support wurde im März 2014 eingeführt und wird nur für Linux-Stacks verwendet. Er läuft mit Ruby 2.0.0 und verwendet den Chef-Client im lokalen Modus. Rezepte können so die Chef-Suchfunktion und Data Bags verwenden. Der Support hat einige Einschränkungen, die in Implementieren von Rezepten: Chef 11.10 beschrieben sind. Sie können jedoch viele Community-Rezeptbücher ohne Änderung ausführen. Sie können auch Berkshelf verwenden, um Ihre Rezeptbuchabhängigkeiten zu verwalten. Die unterstützten Berkshelf-Versionen hängen vom Betriebssystem ab. Weitere Informationen finden Sie unter AWS OpsWorks Stacks-Betriebssysteme. Sie können keine CentOS-Stacks erstellen, die Chef 11.10 verwenden.

Chef 11.4

Der Chef 11.4-Support wurde im Juli 2013 eingeführt und wird nur für Linux-Stacks verwendet. Er läuft mit Ruby 1.8.7 und verwendet chef-solo. Hiermit werden die Chef-Suchfunktion oder Data Bags nicht unterstützt. Mit AWS OpsWorks Stacks können Sie häufig Community-Kochbücher verwenden, die von diesen Funktionen abhängen, aber Sie müssen sie wie unter beschrieben ändern. Migrieren auf eine neue Chef-Version Sie können keine CentOS-Stacks erstellen, die

Chef 11.4 verwenden. Chef 11.4-Stacks werden auf regionalen Endpunkten außerhalb der Region USA Ost (Nord-Virginia) nicht unterstützt.

Chef 0.9

Chef 0.9 wird nur für Linux-Stacks verwendet und nicht mehr unterstützt. Beachten Sie die folgenden Informationen:

- Mit der Konsole können Sie keinen neuen Chef 0.9-Stack erstellen.

  Sie müssen die Befehlszeilen-Schnittstelle (CLI) bzw. eine API verwenden oder einen Stack mit einer anderen Chef-Version erstellen und anschließend die Stack-Konfiguration bearbeiten.
- Neue AWS OpsWorks Stacks-Funktionen sind für Chef 0.9-Stacks nicht verfügbar.
- Neue Betriebssystemversionen bieten nur begrenzten Support für Chef 0.9-Stacks.

  Insbesondere Amazon Linux 2014.09 und spätere Versionen unterstützen Chef 0.9-Stacks mit Rails App Server-Layern, die von Ruby 1.8.7 abhängen, nicht.
- Neue AWS-Regionen, einschließlich Europa (Frankfurt), unterstützen Chef 0.9-Stacks nicht.

> **Note**
>
> Wir empfehlen, Chef 0.9 nicht für neue Stacks zu verwenden. Sie sollten bestehende Stacks so bald wie möglich auf die neueste Chef-Version migrieren.

Wenn Sie Community-Kochbücher mit AWS OpsWorks Stacks verwenden möchten, empfehlen wir Ihnen, Chef 12 für neue Linux-Stacks anzugeben und Ihre vorhandenen Linux-Stacks auf Chef 12 zu migrieren. Sie können die AWS OpsWorks Stacks-Konsole, API oder CLI verwenden, um Ihre vorhandenen Stacks auf eine neuere Chef-Version zu migrieren. Weitere Informationen finden Sie unter Migrieren auf eine neue Chef-Version.

Themen

- Implementierung von Rezepten für Chef 12.2 Stacks
- Implementieren von Rezepten für Chef 12-Stacks
- Implementieren von Rezepten für Chef 11.10-Stacks
- Implementieren von Rezepten für Chef 11.4-Stacks
- Migrieren eines vorhandenen Linux-Stacks auf eine neue Chef-Version

Implementierung von Rezepten für Chef 12.2 Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Chef 12.2 (derzeit Chef 12.22) steht nur auf Windows-Stacks zur Verfügung, die diese Chef-Version ausführen müssen.

- Rezepte müssen Windows-spezifische Attribute und Ressourcen für einige Zwecke verwenden.

  Weitere Informationen finden Sie unter Chef für Microsoft Windows.
- Da Chef-Läufe Ruby 2.3.6 verwenden, können Ihre Rezepte die neue Ruby-Syntax nutzen.
- Rezepte können die Chef-Suchfunktion und Data Bags verwenden.

  Chef 12.2 Stacks können viele Community-Kochbücher ohne Änderung verwenden. Weitere Informationen finden Sie unter Verwenden der Chef-Suchfunktion und Verwenden von Data Bags.
- Die meisten der in AWS OpsWorks Referenz für Stacks Data Bag und Integrierte Rezeptbuchattribute beschriebenen Stack-Konfigurations- und Bereitstellungsattribute sind für Windows-Rezepte verfügbar.

  Sie können diese Attributwerte mit der Chef-Suchfunktion abrufen. Ein Beispiel finden Sie unter Abrufen von Attributwerten mit der Chef-Suche. Eine Liste von Attributen finden Sie unter AWS OpsWorks Referenz für Stacks Data Bag.

Implementieren von Rezepten für Chef 12-Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Chef 12-Stacks bieten die folgenden Vorteile gegenüber Chef 11.10-Stacks:

- Da Chef-Läufe Ruby 2.1.6 verwenden, können Ihre Rezepte die neue Ruby-Syntax nutzen.
- Chef 12-Stacks können noch mehr Community-Rezeptbücher ohne Änderung verwenden. Ohne integrierte Rezeptbücher kann es nicht mehr zu Namenskonflikten zwischen integrierten und benutzerdefinierten Rezeptbüchern kommen.
- Sie sind nicht mehr auf die Berkshelf-Versionen beschränkt, für die AWS OpsWorks Stacks vorgefertigte Pakete bereitgestellt hat. Berkshelf ist in Chef 12 nicht mehr auf AWS OpsWorks Stacks-Instanzen installiert. Stattdessen können Sie eine beliebige Berkshelf-Version auf Ihrer lokalen Workstation verwenden.
- Es gibt jetzt eine klare Trennung zwischen den integrierten Kochbüchern, die AWS OpsWorks Stacks mit Chef 12 (Elastic Load Balancing, Amazon RDS und Amazon ECS) bereitstellt, und benutzerdefinierten Kochbüchern. Dadurch ist die Fehlerbehebung bei fehlgeschlagenen Chef-Läufen einfacher.

Implementieren von Rezepten für Chef 11.10-Stacks

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Chef 11.10-Stacks bieten die folgenden Vorteile gegenüber Chef 11.4-Stacks:

- Da Chef-Läufe Ruby 2.0.0 verwenden, können Ihre Rezepte die neue Ruby-Syntax nutzen.
- Rezepte können die Chef-Suchfunktion und Data Bags verwenden.

  Chef 11.10-Stacks können viele Community-Rezeptbücher ohne Änderung nutzen.
- Sie können Berkshelf zum Verwalten von Rezeptbüchern verwenden.

Berkshelf bietet eine flexiblere Möglichkeit zum Verwalten Ihrer benutzerdefinierten Rezeptbücher und zum Verwenden von Community-Rezeptbüchern in einem Stack.

- Rezeptbücher müssen Abhängigkeiten in `metadata.rb` deklarieren.

  Wenn Ihr Rezeptbuch von einem anderen Rezeptbuch abhängt, müssen Sie diese Abhängigkeit in die Datei `metadata.rb` Ihres Rezeptbuchs aufnehmen. Wenn Ihr Rezeptbuch beispielsweise ein Rezept mit einem Statement wie `include_recipe anothercookbook::somerecipe` enthält, muss Ihre Rezeptbuch-Datei `metadata.rb` die folgende Zeile enthalten: `depends "anothercookbook"`.
- AWS OpsWorks Stacks installiert einen MySQL-Client nur dann auf den Instanzen eines Stacks, wenn der Stack eine MySQL-Schicht enthält.
- AWS OpsWorks Stacks installiert einen Ganglia-Client nur dann auf den Instanzen eines Stacks, wenn der Stack eine Ganglia-Schicht enthält.
- Wenn eine Bereitstellung `bundle install` ausführt und bei der Installation ein Fehler auftritt, kann die Bereitstellung auch nicht verarbeitet werden.

> **Important**
>
> Verwenden Sie keine Namen der integrierten Rezeptbücher für benutzerdefinierte oder Community-Rezeptbücher. Bei benutzerdefinierten Rezeptbüchern mit demselben Namen wie integrierte Rezeptbücher kann ein Fehler auftreten. Eine vollständige Liste der integrierten Kochbücher, die mit den Stacks Chef 11.10, 11.4 und 0.9 verfügbar sind, finden Sie im Opsworks-Cookbooks-Repository unter. GitHub
>
> Bei Rezeptbüchern mit Nicht-ASCII-Zeichen, die in Chef 0.9- und 11.4-Stacks erfolgreich ausgeführt werden, kann auf einem Chef 11.10-Stack ein Fehler auftreten. Der Grund ist, dass Chef 11.10-Stacks Ruby 2.0.0 für Chef-Ausführungen verwenden, bei dem die Kodierungsrichtlinien viel strenger sind als bei Ruby 1.8.7. Um sicherzustellen, dass diese Rezeptbücher auf Chef 11.10-Stacks erfolgreich ausgeführt werden, sollte jede Datei mit Nicht-ASCII-Zeichen oben mit einem Kommentar, der einen Hinweis zur Kodierung enthält, versehen sein. Für die UTF-8-Kodierung würde der Kommentar z. B. `# encoding: UTF-8` lauten. Weitere Informationen zur Ruby 2.0.0-Kodierung finden Sie unter Kodierung.

Themen

- Installation und Vorrang von Rezeptbüchern

- Verwenden der Chef-Suchfunktion
- Verwenden von Data Bags
- Verwenden von Berkshelf

Installation und Vorrang von Rezeptbüchern

Das Verfahren zur Installation von AWS OpsWorks Stacks-Kochbüchern funktioniert für Chef 11.10-Stacks etwas anders als für frühere Chef-Versionen. Bei Chef 11.10-Stacks werden die integrierten, benutzerdefinierten und Berkshelf-Kochbücher nach der Installation von AWS OpsWorks Stacks in der folgenden Reihenfolge zu einem gemeinsamen Verzeichnis zusammengeführt:

1. Integrierte Rezeptbücher.
2. Berkshelf-Rezeptbücher, sofern vorhanden.
3. Benutzerdefinierte Rezeptbücher, sofern vorhanden.

Wenn AWS OpsWorks Stacks diese Zusammenführung durchführt, kopiert es den gesamten Inhalt der Verzeichnisse, einschließlich der Rezepte. Wenn Duplikate vorhanden sind, gelten die folgenden Regeln:

- Der Inhalt der Berkshelf-Rezeptbücher hat Vorrang vor den integrierten Rezeptbüchern.
- Der Inhalt der benutzerdefinierten Rezeptbücher hat Vorrang vor den Berkshelf-Rezeptbüchern.

Um zu veranschaulichen, wie dieser Vorgang funktioniert, sehen Sie sich das folgende Szenario an, in dem alle drei Rezeptbuchverzeichnisse ein Rezeptbuch mit dem Namen `mycookbook` enthalten:

- Integrierte Kochbücher — `mycookbook` enthält eine Attributdatei mit dem Namen`someattributes.rb`, eine Vorlagendatei mit dem Namen `sometemplate.erb` und ein Rezept mit dem Namen. `somerecipe.rb`
- Berkshelf-Kochbücher — beinhaltet und. `mycookbook` `sometemplate.erb` `somerecipe.rb`
- Benutzerdefinierte Kochbücher — beinhaltet. `mycookbook` `somerecipe.rb`

Das zusammengeführte Rezeptbuch enthält Folgendes:

- `someattributes.rb` aus dem integrierten Rezeptbuch.
- `sometemplate.erb` aus dem Berkshelf-Rezeptbuch.

- `somerecipe.rb` aus dem benutzerdefinierten Rezeptbuch.

> **Important**
>
> Sie sollten Ihren Chef 11.10-Stack nicht anpassen, indem Sie ein komplettes integriertes Rezeptbuch in Ihr Repository kopieren und dann Teile des Rezeptbuchs ändern. Dabei wird das gesamte integrierte Rezeptbuch, einschließlich der Rezepte, überschrieben. Wenn AWS OpsWorks Stacks dieses Kochbuch aktualisiert, kann dein Stack nicht von diesen Updates profitieren, es sei denn, du aktualisierst deine private Kopie manuell. Weitere Informationen zum Anpassen von Stacks finden Sie unter Stacks anpassen AWS OpsWorks.

Verwenden der Chef-Suchfunktion

Sie können die Chef-`search`-Methode in Ihren Rezepten verwenden, um Stack-Daten abzufragen. Sie verwenden dieselbe Syntax wie für den Chef-Server, aber AWS OpsWorks Stacks bezieht die Daten vom lokalen Knotenobjekt, anstatt einen Chef-Server abzufragen. Diese Daten umfassen Folgendes:

- Die Stack-Konfigurations- und Bereitstellungsattribute der Instance.
- Die Attribute aus den Attributdateien der integrierten und benutzerdefinierten Rezeptbücher der Instance.
- Von Ohai gesammelte Systemdaten.

Die Stack-Konfiguration und die Bereitstellungsattribute enthalten die meisten Informationen, die Rezepte normalerweise durch Suchen abrufen, einschließlich Daten wie Hostnamen und IP-Adressen für jede Online-Instanz im Stack. AWS OpsWorks Stacks aktualisiert diese Attribute für jedes Lebenszyklusereignis, wodurch sichergestellt wird, dass sie den aktuellen Stack-Status genau wiedergeben. Das bedeutet, dass Sie suchabhängige Community-Rezepte in Ihrem Stack häufig ohne Änderung verwenden können. Die Suchmethode gibt weiterhin die entsprechenden Daten zurück. Sie stammen nur aus den Stack-Konfigurations- und Bereitstellungsattributen statt von einem Server.

Die Haupteinschränkung der AWS OpsWorks Stacks-Suche besteht darin, dass nur die Daten im lokalen Knotenobjekt verarbeitet werden, insbesondere die Stack-Konfiguration und die Bereitstellungsattribute. Aus diesem Grund sind die folgenden Arten von Daten über die Suche möglicherweise nicht verfügbar:

- Lokal definierte Attribute auf anderen Instances.

  Wenn ein Rezept ein Attribut lokal definiert, werden diese Informationen nicht an den AWS OpsWorks Stacks-Dienst zurückgemeldet, sodass Sie nicht von anderen Instanzen aus auf diese Daten zugreifen können, indem Sie die Suche verwenden.

- Benutzerdefinierte `deploy`-Attribute.

  Sie können das benutzerdefinierte JSON-Objekt bei der Bereitstellung einer App angeben. Die entsprechenden Attribute werden auf den Instances des Stacks für diese Bereitstellung installiert. Wenn Sie die Bereitstellung jedoch nur für ausgewählte Instances vornehmen, werden die Attribute nur auf diesen Instances installiert. Abfragen für diese benutzerdefinierten JSON-Attribute schlagen auf allen anderen Instances fehl. Darüber hinaus sind die benutzerdefinierten Attribute in der Stack-Konfigurations- und Bereitstellungs-JSON nur für die jeweilige Bereitstellung enthalten. Auf die Attribute kann erst zugegriffen werden, wenn das nächste Lebenszyklusereignis eine neue Reihe von Stack-Konfigurations- und Bereitstellungsattributen installiert. Hinweis: Wenn Sie ein benutzerdefiniertes JSON-Objekt für den Stack angeben, werden die Attribute auf allen Instances für jedes Lebenszyklusereignis installiert und können immer über die Suche gefunden werden.

- Ohai-Daten von anderen Instances.

  Das Chef-Ohai-Tool ruft eine Vielzahl von Daten auf einer Instance ab und fügt sie dem Knotenobjekt hinzu. Diese Daten werden lokal gespeichert und nicht dem AWS OpsWorks Stacks-Service gemeldet, sodass die Suchfunktion nicht auf Ohai-Daten von anderen Instances zugreifen kann. Einige dieser Daten können jedoch in die Stack-Konfigurations- und Bereitstellungsattribute aufgenommen werden.

- Offline-Instances.

  Die Stack-Konfigurations- und Bereitstellungsattribute enthalten nur Daten für Online-Instances.

Der folgende Rezeptauszug zeigt, wie die private IP-Adresse einer PHP-Layer-Instance mit der Suchfunktion abgerufen wird.

```
appserver = search(:node, "role:php-app").first
Chef::Log.info("The private IP is '#{appserver[:private_ip]}'")
```

> **Note**
>
> Wenn AWS OpsWorks Stacks dem Node-Objekt die Stack-Konfiguration und die Bereitstellungsattribute hinzufügt, werden tatsächlich zwei Sätze von Layer-Attributen mit jeweils denselben Daten erstellt. Ein Satz befindet sich im `layers` Namespace, in dem AWS OpsWorks Stacks die Daten speichert. Die andere Gruppe wird im `role`-Namespace abgelegt, d. h. so speichert der Chef Server die entsprechenden Daten. Der Zweck des `role` Namespace besteht darin, dass Suchcode, der für den Chef-Server implementiert wurde, auf einer AWS OpsWorks Stacks-Instanz ausgeführt werden kann. Wenn Sie Code speziell für AWS OpsWorks Stacks schreiben, könnten Sie entweder `layers:php-app` oder `role:php-app` im vorherigen Beispiel verwenden und `search` würden dasselbe Ergebnis zurückgeben.

Verwenden von Data Bags

Sie können die Chef-`data_bag_item`-Methode in Ihren Rezepten für die Abfrage von Informationen in einem Data Bag verwenden. Sie verwenden die gleiche Syntax wie für einen Chef-Server. AWS OpsWorks Stacks erhält die Daten allerdings aus den Stack-Konfigurations- und Bereitstellungsattributen der Instance. AWS OpsWorks Stacks unterstützt derzeit jedoch keine Chef-Umgebungen und kehrt daher `node.chef_environment` immer zurück. `_default`

Sie erstellen ein Data Bag mit einer benutzerdefinierten JSON, um dem `[:opsworks][:data_bags]`-Attribut ein oder mehrere Attribute hinzuzufügen. Das folgende Beispiel zeigt das allgemeine Format zum Erstellen eines Data Bags in einer benutzerdefinierten JSON.

> **Note**
>
> Sie können kein Data Bag erstellen, indem Sie es Ihrem Rezeptbuch-Repository hinzufügen. Sie müssen ein benutzerdefiniertes JSON-Objekt verwenden.

```
{
  "opsworks": {
    "data_bags": {
      "bag_name1": {
        "item_name1: {
          "key1" : “value1”,
```

```
          "key2" : “value2”,
          ...
        }
      },
      "bag_name2": {
        "item_name1": {
          "key1" : “value1”,
          "key2" : “value2”,
          ...
        }
      },
      ...
    }
  }
}
```

Sie geben das benutzerdefinierte JSON-Objekt normalerweise für den Stack an, der die benutzerdefinierten Attribute auf jede Instance für jedes folgende Lebenszyklusereignis installiert. Sie können ein benutzerdefiniertes JSON-Objekt auch beim Bereitstellen einer Anwendung angeben. Diese Attribute werden jedoch nur für diese Bereitstellung installiert und zwar möglicherweise nur für eine bestimmte Gruppe von Instances. Weitere Informationen finden Sie unter Bereitstellen von Anwendungen.

Das folgende Beispiel zeigt, wie ein benutzerdefiniertes JSON-Objekt ein Data Bag mit dem Namen `myapp` erstellt. Die JSON verfügt über ein Element, `mysql`, mit zwei Schlüssel-Wert-Paaren.

```
{ "opsworks": {
    "data_bags": {
      "myapp": {
        "mysql": {
          "username": "default-user",
          "password": "default-pass"
        }
      }
    }
  }
}
```

Um die Daten in Ihrem Rezept zu verwenden, können Sie `data_bag_item` aufrufen und das Data Bag und die Wertenamen übergeben, wie im folgenden Auszug dargestellt.

```
mything = data_bag_item("myapp", "mysql")
Chef::Log.info("The username is '#{mything['username']}' ")
```

Zum Ändern der Daten im Data Bag modifizieren Sie nur das benutzerdefinierte JSON-Objekt. Die Installation erfolgt dann auf den Instances des Stacks für das nächste Lebenszyklusereignis.

Verwenden von Berkshelf

Mit Chef 0.9- und Chef 11.4-Stacks können Sie nur ein benutzerdefiniertes Rezeptbuch-Repository installieren. Mit Chef 11.10-Stacks können Sie Berkshelf für die Verwaltung Ihrer Rezeptbücher und deren Abhängigkeiten verwenden. So können Sie Rezeptbücher aus mehreren Repositorys installieren. (Weitere Informationen finden Sie unter Lokales Verpacken von Rezeptbuch-Abhängigkeiten.) Insbesondere können Sie mit Berkshelf AWS OpsWorks Stacks-kompatible Community-Kochbücher direkt aus ihren Repositorys installieren, anstatt sie in Ihr benutzerdefiniertes Kochbuch-Repository kopieren zu müssen. Die unterstützten Berkshelf-Versionen hängen vom Betriebssystem ab. Weitere Informationen finden Sie unter AWS OpsWorks Stacks-Betriebssysteme.

Zum Verwenden von Berkshelf müssen Sie eine Aktivierung vornehmen, wie in Installieren von benutzerdefinierten Rezeptbüchern beschrieben. Nehmen Sie dann eine `Berksfile`-Datei in das Stammverzeichnis Ihres Rezeptbuch-Repositorys auf, das die zu installierenden Rezeptbücher festlegt.

Um eine externe Rezeptbuchquelle in einer Berksfile-Datei festzulegen, geben Sie ein Quellattribut oben in der Datei an, das die Standard-Repository-URL festlegt. Berkshelf sucht die Rezeptbücher in den Quell-URLs, es sei denn, Sie geben ein Repository explizit an. Fügen Sie dann eine Zeile für die einzelnen Rezeptbücher hinzu, die Sie installieren möchten. Verwenden Sie dazu folgendes Format:

```
cookbook 'cookbook_name', ['>= cookbook_version'], [cookbook_options]
```

Die Felder nach `cookbook` geben das jeweilige Rezeptbuch an.

- *`cookbook_name`* — (Erforderlich) Gibt den Namen des Kochbuches an.

  Wenn Sie keine weiteren Felder angeben, installiert Berkshelf das Rezeptbuch mit den angegebenen Quell-URLs.
- *`cookbook_version – (Optional) Gibt die Version`* oder die Versionen des Kochbuchs an.

Sie können ein Präfix festlegen, wie z. B. = oder >=, um eine bestimmte Version oder eine Reihe gültiger Versionen anzugeben. Wenn Sie keine Version angeben, installiert Berkshelf die aktuelle Version.

- *`cookbook_options`* — (Optional) Das letzte Feld ist ein Hash, der ein oder mehrere Schlüssel-Wert-Paare enthält, die Optionen wie den Speicherort des Repositorys angeben.

  Sie können beispielsweise einen `git`-Schlüssel angeben, um auf ein bestimmtes Git-Repository zu verweisen, und einen `tag`-Schlüssel für eine bestimmte Repository-Branch festlegen. Die Angabe der Repository-Branch ist in der Regel der beste Weg, um sicherzustellen, dass Sie Ihr bevorzugtes Rezeptbuch installieren.

> **Important**
>
> Deklarieren Sie keine Rezeptbücher, indem Sie eine `metadata`-Zeile in Ihrer Berksfile-Datei einfügen und die Rezeptbuchabhängigkeiten in der Datei `metadata.rb` deklarieren. Damit dies einwandfrei funktioniert, müssen beide Dateien im selben Verzeichnis gespeichert sein. Bei AWS OpsWorks Stacks muss sich das Berksfile im Stammverzeichnis des Repositorys befinden, aber die `metadata.rb` Dateien müssen sich in ihren jeweiligen Kochbuchverzeichnissen befinden. Deklarieren Sie stattdessen externe Rezeptbücher explizit in der Berksfile-Datei.

Es folgt ein Beispiel für eine Berksfile-Datei, das die verschiedenen Möglichkeiten zum Angeben von Rezeptbüchern veranschaulicht. Weitere Informationen zum Erstellen einer Berksfile-Datei finden Sie unter 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'
```

Mit dieser Datei werden die folgenden Rezeptbücher installiert:

- Die aktuelle Version von `apt` aus dem Repository der Community-Rezeptbücher.
- Die aktuelle Version von `bluepill` der Community-Rezeptbücher, sofern es sich um Version 2.3.1 oder höher handelt.
- Die aktuelle Version von `ark` aus einem angegebenen Repository.

  Die URL für dieses Beispiel ist für ein öffentliches Community-Kochbuch-Repository aktiviert GitHub, aber Sie können Kochbücher aus anderen Repositorys installieren, auch aus privaten Repositorys. Weitere Informationen finden Sie unter Berkshelf.
- Das `build-essential`-Rezeptbuch aus der v1.4.2-Branch des angegebenen Repositorys.

Ein benutzerdefiniertes Rezeptbuch-Repository kann zusätzlich zu einer Berksfile-Datei benutzerdefinierte Rezeptbücher enthalten. In diesem Fall installiert AWS OpsWorks Stacks beide Gruppen von Kochbüchern, was bedeutet, dass eine Instanz über bis zu drei Kochbuch-Repositorys verfügen kann.

- Die integrierten Rezeptbücher werden im Verzeichnis `/opt/aws/opsworks/current/cookbooks` installiert.
- Wenn Ihr benutzerdefiniertes Rezeptbuch-Repository Rezeptbücher enthält, werden sie in das Verzeichnis `/opt/aws/opsworks/current/site-cookbooks` installiert.
- Wenn Sie Berkshelf aktiviert haben und Ihr benutzerdefiniertes Rezeptbuch-Repository eine Berksfile-Datei enthält, werden die angegebenen Rezeptbücher im Verzeichnis `/opt/aws/opsworks/current/berkshelf-cookbooks` installiert.

Die integrierten Kochbücher und Ihre benutzerdefinierten Kochbücher werden während der Einrichtung auf jeder Instanz installiert und anschließend nicht aktualisiert, es sei denn, Sie führen den Stack-Befehl „Benutzerdefinierte Kochbücher aktualisieren“ manuell aus. AWS OpsWorks Stacks läuft `berks install` bei jedem Koch-Lauf, sodass Ihre Berkshelf-Kochbücher für jedes Lebenszyklusereignis gemäß den folgenden Regeln aktualisiert werden:

- Bei einer neuen Version im Repository wird mit diesem Vorgang das Rezeptbuch aus dem Repository aktualisiert.
- Andernfalls aktualisiert dieser Vorgang die Berkshelf-Rezeptbücher aus einem lokalen Cache.

> **Note**
>
> Mit dem Vorgang werden die Berkshelf-Rezeptbücher überschrieben. Wenn Sie die lokalen Kopien der Rezeptbücher geändert haben, werden die Änderungen hiermit überschrieben. Weitere Informationen finden Sie unter Berkshelf.

Sie können Ihre Berkshelf-Rezeptbücher auch aktualisieren, indem Sie den Stack-Befehl Benutzerdefinierte Rezeptbücher aktualisieren ausführen. Mit diesem Befehl werden sowohl die Berkshelf-Rezeptbücher als auch Ihre benutzerdefinierten Rezeptbücher aktualisiert.

Implementieren von Rezepten für Chef 11.4-Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Important**
>
> Verwenden Sie keine Namen der integrierten Rezeptbücher für benutzerdefinierte oder Community-Rezeptbücher. Bei benutzerdefinierten Rezeptbüchern mit demselben Namen wie integrierte Rezeptbücher kann ein Fehler auftreten. Eine vollständige Liste der integrierten Kochbücher, die mit den Chef-Stacks 11.10, 11.4 und 0.9 verfügbar sind, finden Sie im opsworks-cookbooks-Repository unter. GitHub

Die primäre Einschränkung von Chef 11.4-Stacks besteht darin, dass Rezepte weder die Chef-Suchfunktion noch Data Bags verwenden können. AWS OpsWorks Stacks installiert jedoch Stackkonfigurations- und Bereitstellungsattribute auf jeder Instanz, die viele der Informationen enthalten, die Sie mit der Suche erhalten würden, einschließlich der folgenden:

- Benutzerdefinierte Daten von der Konsole, wie z. B. Host- oder App-Namen.

- Vom AWS OpsWorks Stacks-Dienst generierte Stack-Konfigurationsdaten, wie z. B. die Ebenen, Apps und Instanzen des Stacks, sowie Details zu jeder Instanz, wie z. B. die IP-Adresse.
- Benutzerdefinierte JSON-Attribute, die vom Benutzer bereitgestellte Daten enthalten und nahezu denselben Zweck erfüllen können wie Data Bags.

AWS OpsWorks Stacks installiert für jedes Lebenszyklusereignis eine aktuelle Version der Stack-Konfiguration und der Bereitstellungsattribute auf jeder Instanz, bevor der Chef-Lauf des Events gestartet wird. Die Daten werden den Rezepten mit der Standardsyntax `node[:attribute][:child_attribute][...]` zur Verfügung gestellt. Die Stack-Konfigurations- und Bereitstellungsattribute enthalten z. B. den Stack-Namen `node[:opsworks][:stack][:name]`.

Der folgende Auszug aus einem der integrierten Rezepte erhält den Stack-Namen und verwendet ihn zum Erstellen einer Konfigurationsdatei.

```
template '/etc/ganglia/gmetad.conf' do
  source 'gmetad.conf.erb'
  mode '0644'
  variables :stack_name => node[:opsworks][:stack][:name]
  notifies :restart, "service[gmetad]"
end
```

Viele der Stack-Konfigurations- und Bereitstellungsattributwerte enthalten mehrere Attribute. Sie müssen diese Attribute schrittweise durchlaufen, um die benötigten Informationen zu erhalten. Das folgende Beispiel zeigt einen Auszug aus den Stack-Konfigurations- und Bereitstellungsattributen, die der Einfachheit halber als JSON-Objekt dargestellt werden. Es enthält ein Top-Level-Attribut, `deploy`, mit einem Attribut für jede App des Stacks, die mit dem Kurznamen der App bezeichnet wird.

```
{
  ...
  "deploy": {
    "app1_shortname": {
      "document_root": "app1_root",
      "deploy_to": "deploy_directory",
      "application_type": "php",
      ...
    },
```

```
    "app2_shortname": {
      "document_root": "app2_root",
      ...
    }
  },
  ...
}
```

Jedes App-Attribut enthält eine Gruppe von Attributen, die die Merkmale der Anwendung angeben. Das `deploy_to`-Attribut stellt z. B. das Bereitstellungsverzeichnis der App dar. Mit dem folgenden Auszug werden Benutzer, Gruppe und Pfad für das Bereitstellungsverzeichnis der einzelnen Apps festgelegt.

```
node[:deploy].each do |application, deploy|
  opsworks_deploy_dir do
    user deploy[:user]
    group deploy[:group]
    path deploy[:deploy_to]
  end
  ...
end
```

Weitere Informationen zu den Stack-Konfigurations- und Bereitstellungsattributen finden Sie unter Stacks anpassen AWS OpsWorks. Weitere Informationen zu den Bereitstellungsverzeichnissen finden Sie unter Bereitstellungsrezepte.

Chef 11.4-Stacks unterstützen keine Data Bags. Sie können den Stack-Konfigurations- und Bereitstellungsattributen jedoch beliebige Daten hinzufügen, indem Sie eine benutzerdefinierte JSON angeben. Ihre Rezepte können dann mit der standardmäßigen Chef-Knotensyntax auf die Daten zugreifen. Weitere Informationen finden Sie unter Nutzen eines benutzerdefinierten JSON-Objekts.

Wenn Sie die Funktionalität einer verschlüsselten Datentasche benötigen, besteht eine Möglichkeit darin, vertrauliche Attribute an einem sicheren Ort wie einem privaten Amazon S3 S3-Bucket zu speichern. Ihre Rezepte können dann das AWS Ruby SDK verwenden, das auf allen AWS OpsWorks Stacks-Instances installiert ist, um die Daten aus dem Bucket herunterzuladen.

> **Note**
>
> Jede AWS OpsWorks Stacks-Instance hat ein Instance-Profil. Die zugehörige IAM-Rolle gibt an, auf welche AWS-Ressourcen von Anwendungen zugegriffen werden kann, die auf der

Instance ausgeführt werden. Damit Ihre Rezepte auf einen Amazon S3 S3-Bucket zugreifen können, muss die Richtlinie der Rolle eine Aussage ähnlich der folgenden enthalten, die die Berechtigung zum Abrufen von Dateien aus einem bestimmten Bucket erteilt.

```
"Action": ["s3:GetObject"],
"Effect": "Allow",
"Resource": "arn:aws:s3:::yourbucketname/*",
```

Weitere Informationen zu Instance-Profilen finden Sie unter Festlegen von Berechtigungen für Apps auf EC2-Instances.

Migrieren eines vorhandenen Linux-Stacks auf eine neue Chef-Version

Important

Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können die AWS OpsWorks Stacks-Konsole, die API oder die CLI verwenden, um Ihre Linux-Stacks auf eine neuere Chef-Version zu migrieren. Für Ihre Rezepte ist jedoch möglicherweise eine Änderung erforderlich, damit sie mit der neueren Version kompatibel sind. Beachten Sie im Vorfeld der Migration eines Stacks die folgenden Hinweise.

- Sie können die AWS OpsWorks Stacks-Stack-Versionen nicht von Chef 11 auf Chef 12 ändern, indem Sie den Stack bearbeiten oder klonen. Ein Upgrade der Chef-Hauptversion kann mit dem in diesem Abschnitt beschriebenen Verfahren nicht durchgeführt werden. Weitere Informationen zur Umstellung von Chef 11.10 auf Chef 12 finden Sie unter Implementieren von Rezepten: Chef 12.
- Die Umstellung von einer Chef-Version auf eine andere beinhaltet eine Reihe von Änderungen, die zum Teil grundlegend sind.

  Weitere Informationen zur Umstellung von Chef 0.9 auf Chef 11.4 finden Sie unter Migrieren auf eine neue Chef-Version. Weitere Informationen zur Umstellung von Chef 11.4 auf Chef 11.10

finden Sie unter Implementieren von Rezepten: Chef 11.10. Weitere Informationen zur Umstellung von Chef 11.10 auf Chef 12 finden Sie unter Implementieren von Rezepten: Chef 12.

- Chef-Läufe verwenden eine andere Ruby-Version auf Chef 0.9- und Chef 11.4-Stacks (Ruby 1.8.7), Chef 11.10-Stacks (Ruby 2.0.0) und Chef 12-Stacks (Ruby-2.1.6).

  Weitere Informationen finden Sie unter Ruby-Versionen.

- Chef 11.10-Stacks nehmen die Rezeptbuchinstallation von Chef 0.9- oder Chef 11.4-Stacks unterschiedlich vor.

  Dieser Unterschied kann zu Problemen führen, wenn Sie Stacks mit benutzerdefinierten Rezeptbüchern auf Chef 11.10 migrieren. Weitere Informationen finden Sie unter Installation und Vorrang von Rezeptbüchern.

Die folgenden Richtlinien werden für das Migrieren eines Chef-Stacks auf eine neuere Chef-Version empfohlen:

Migrieren eines Stacks auf eine neuere Chef-Version

1. Klonen Sie Ihren Produktions-Stack. Klicken Sie auf der Seite Clone Stack auf Advanced >>, um den Abschnitt Configuration Management anzuzeigen, und ändern Sie Chef version auf die nächste höhere Version.

   > **Note**
   >
   > Wenn Sie mit einem Chef 0.9-Stack beginnen, können Sie kein Upgrade direkt auf Chef 11.10 durchführen. Sie müssen zunächst ein Upgrade auf Chef 11.4 vornehmen. Wenn Sie Ihren Stack auf Chef 11.10 migrieren möchten, bevor Sie Ihre Rezepte testen, warten Sie 20 Minuten, bis die Aktualisierung ausgeführt wird, und führen Sie dann ein Upgrade des Stacks von 11.4 auf 11.10 durch.

2. Fügen Sie den Layern Instances hinzu und testen Sie die geklonten Stack-Anwendungen und Rezeptbücher auf einem Test- oder Staging-System. Weitere Informationen finden Sie unter All about Chef ....
3. Wenn die Testergebnisse zufriedenstellend sind, führen Sie einen der folgenden Schritte aus:

   - Wenn dies die gewünschte Chef-Version ist, können Sie den geklonten Stack als Produktions-Stack verwenden oder die Chef-Version auf Ihrem Produktions-Stack zurücksetzen.

- Wenn Sie einen Chef 0.9-Stack auf Chef 11.10 in zwei Phasen migrieren, wiederholen Sie den Prozess, um den Stack von Chef 11.4 auf Chef 11.10 zu migrieren.

> **Note**
>
> Wenn Sie Rezepte testen, können Sie über SSH eine Verbindung mit der Instance herstellen und dann den Instance-Agenten-CLI-Befehl run_command zum Ausführen der mit den verschiedenen Lebenszyklusereignissen verbundenen Rezepte ausführen. Die Agenten-CLI ist besonders nützlich zum Testen von Einrichtungsrezepten, da Sie sie sogar verwenden können, wenn die Einrichtung fehlschlägt und die Instance nicht online ist. Sie können auch den Setup-Stack-Befehl verwenden, um Einrichtungsrezepte neu zu starten. Dieser Befehl ist jedoch nur verfügbar, wenn die Einrichtung erfolgreich war und die Instance online ist.

Es ist möglich, einen laufenden Stack auf eine neue Chef-Version zu aktualisieren.

Aktualisieren eines laufenden Stacks auf eine neue Chef-Version

1. Bearbeiten Sie den Stack, um die Stack-Einstellung Chef version zu ändern.
2. Speichern Sie die neuen Einstellungen und warten Sie, bis AWS OpsWorks Stacks die Instanzen aktualisiert hat. Dies dauert normalerweise 15 bis 20 Minuten.

> **Important**
>
> AWS OpsWorks Stacks synchronisiert das Chef-Versionsupdate nicht mit Lebenszyklusereignissen. Wenn Sie die Chef-Version auf einem Produktions-Stack aktualisieren möchten, müssen Sie sicherstellen, dass die Aktualisierung abgeschlossen ist, bevor das nächste Lebenszyklusereignis eintritt. Wenn ein Ereignis eintritt — in der Regel ein Deploy- oder Configure-Ereignis — aktualisiert der Instance-Agent Ihre benutzerdefinierten Kochbücher und führt die dem Ereignis zugewiesenen Rezepte aus, unabhängig davon, ob das Versionsupdate abgeschlossen ist oder nicht. Es gibt keine direkte Methode, um zu bestimmen, ob die Versionsaktualisierung abgeschlossen wurde. In den Bereitstellungsprotokollen ist jedoch die Chef-Version enthalten.

Ruby-Versionen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Auf allen Instanzen in einem Linux-Stack ist Ruby installiert. AWS OpsWorks Stacks installiert auf jeder Instanz ein Ruby-Paket, mit dem Chef-Rezepte und der Instanzagent ausgeführt werden. AWS OpsWorks Stacks bestimmt die Ruby-Version anhand der Chef-Version, auf der der Stack ausgeführt wird. Ändern Sie diese Version nicht, da hierdurch der Instance-Agent deaktiviert werden könnte.

AWS OpsWorks Stacks installiert keine Ruby-Anwendung, die auf Windows-Stacks ausführbar ist. Der Chef 12.2-Client wird mit Ruby 2.0.0 p451 geliefert, aber die ausführbare Ruby-Datei wird nicht zur PATH-Umgebungsvariablen der Instanz hinzugefügt. Sie können mit dieser ausführbaren Datei auch den Ruby-Code ausführen, er befindet sich im Verzeichnis `\opscode\chef\embedded\bin\ruby.exe` auf Ihrem Windows-Laufwerk.

Die folgende Tabelle fasst die Ruby-Versionen von Stacks zusammen. AWS OpsWorks Die verfügbaren Ruby-Anwendungsversionen hängen auch vom Betriebssystem der Instance ab. Weitere Informationen dazu und zu den verfügbaren Patch-Versionen finden Sie unter AWS OpsWorks Stacks-Betriebssysteme.

Chef-Version	Chef-Ruby-Version	Verfügbare Ruby-Anwendungsversionen
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	Keine
12.22 (d)	2.3.6	None

(a) Nicht verfügbar für Amazon Linux 2014.09 und höher, Red Hat Enterprise Linux (RHEL) oder Ubuntu 14.04 LTS.

(b) Nur auf Linux-Stacks verfügbar.

(c) Nicht für RHEL verfügbar.

(d) Nur auf Windows-Stacks verfügbar. Hauptversion ist 12.2. Die aktuelle Unterversion ist 12.22.

(e) Deprecation abgeschlossen; Unterstützung ist abgelaufen.

Die Installationsverzeichnisse hängen von der Chef-Version ab:

- Anwendungen verwenden die ausführbare Datei `/usr/local/bin/ruby` für alle Chef-Versionen.
- Bei Chef 0.9 und 11.4 verwenden der Instance-Agent und die Chef-Rezepte die ausführbare Datei `/usr/bin/ruby`.
- Bei Chef 11.10 verwenden der Instance-Agent und die Chef-Rezepte die ausführbare Datei `/opt/aws/opsworks/local/bin/ruby`.

Installieren von benutzerdefinierten Rezeptbüchern

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

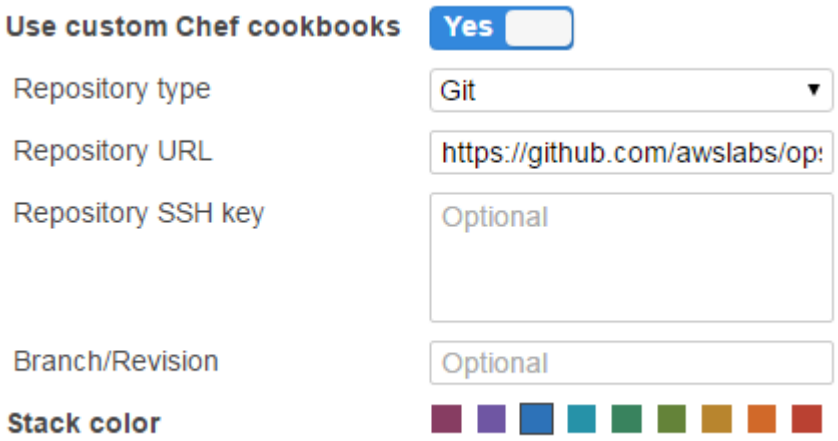
Um auf einem Stack ein benutzerdefiniertes Rezeptbuch zu installieren und zu verwenden, müssen Sie auf dem Stack zunächst benutzerdefinierte Rezepte aktivieren, falls Sie dies noch nicht getan haben. Dann müssen Sie die Repository-URL sowie alle erforderlichen Informationen wie etwa ein Passwort bereitstellen.

> **Important**
>
> Nachdem Sie den Stack so konfiguriert haben, dass er benutzerdefinierte Kochbücher unterstützt, installiert AWS OpsWorks Stacks Ihre Kochbücher beim Start automatisch auf

allen neuen Instanzen. Sie müssen AWS OpsWorks Stacks jedoch ausdrücklich anweisen, neue oder aktualisierte Kochbücher auf allen vorhandenen Instanzen zu installieren, indem Sie den Stack-Befehl Update Custom Cookbooks ausführen. Weitere Informationen finden Sie unter Aktualisieren von benutzerdefinierten Rezeptbüchern. Überprüfen Sie vor dem Aktivieren der Option Use custom Chef cookbooks (Verwenden von benutzerdefinierten Chef-Rezeptbüchern) auf einem Stack, ob die verwendeten benutzerdefinierten und Community-Rezeptbücher die auf dem Stack verwendete Chef-Version unterstützen.

So konfigurieren Sie einen Stack für benutzerdefinierte Rezeptbücher:

1. Klicken Sie auf der Seite des Stacks auf Stack Settings, um die Seite Settings anzuzeigen. Klicken Sie auf Edit, um die Einstellungen zu bearbeiten.
2. Schalten Sie Use custom Chef Cookbooks zu Yes um.



3. Konfigurieren Sie Ihre benutzerdefinierten Rezeptbücher.

Wenn Sie fertig sind, klicken Sie auf Save, um den aktualisierten Stack zu speichern.

Festlegen eines benutzerdefinierten Rezeptbuch-Repositorys

Auf Linux-Stacks können benutzerdefinierte Rezeptbücher aus den folgenden Repository-Typen installiert werden:

- HTTP- oder Amazon S3 S3-Archive.

  Sie können entweder öffentlich oder privat sein, aber Amazon S3 ist in der Regel die bevorzugte Option für ein privates Archiv.

- Mit Git- und Subversion-Repositorys können Sie die Quelle steuern und mehrere Versionen bereitstellen.

Windows Stacks können benutzerdefinierte Kochbücher aus Amazon S3 S3-Archiven und Git-Repositorys installieren.

Alle Repository-Typen haben die folgenden Pflichtfelder.

- Repository-Typ — Der Repository-Typ
- Repository-URL — Die Repository-

AWS OpsWorks Stacks unterstützt öffentlich gehostete Git-Repository-Sites wie GitHubBitbucket sowie privat gehostete Git-Server. Für Git-Repositorys müssen Sie abhängig davon, ob es sich um ein öffentliches oder privates Repository handelt, eines der folgenden URL-Formate verwenden. Dieselben URL-Richtlinien gelten auch für Git-Submodule.

Verwenden Sie für öffentliche Git-Repositorys HTTPS oder Git-Protokolle für schreibgeschützten Zugriff:

- Git schreibgeschützt —. `git://github.com/amazonwebservices/opsworks-example-cookbooks.git`
- HTTPS —. `https://github.com/amazonwebservices/opsworks-example-cookbooks.git`

Für private Git-Repositorys müssen Sie das SSH-Lese-/Schreib-Format verwenden, wie in den folgenden Beispielen dargestellt:

- Github-Repositorien —`git@github.com:`*`project`*`/`*`repository`*.
- Repositorien auf einem Git-Server — *`user`*`@`*`server`*`:`*`project`*`/`*`repository`*

Die übrigen Einstellungen sind abhängig vom Repository-Typ und werden in den folgenden Abschnitten beschrieben.

HTTP-Archiv

Wenn Sie Http Archive für Repository type auswählen, werden zwei zusätzliche Einstellungen angezeigt, die Sie für passwortgeschützte Archive konfigurieren müssen.

- Benutzername — Ihr Benutzername
- Passwort — Ihr Passwort

Amazon S3 S3-Archiv

Wenn Sie S3 Archive (S3-Archiv) für Repository type (Repository-Typ) auswählen, werden die folgenden zusätzlichen, optionalen Einstellungen angezeigt. AWS OpsWorks Stacks kann mithilfe von Amazon EC2 EC2-Rollen (Host Operating System Manager-Authentifizierung) auf Ihr Repository zugreifen, unabhängig davon, ob Sie die AWS OpsWorks Stacks-API oder die Stacks-Konsole verwenden.

- Zugriffsschlüssel-ID — Eine AWS-Zugriffsschlüssel-ID, z. B. AKIAIOSFODNN7EXAMPLE.
- Geheimer Zugriffsschlüssel — Der entsprechende geheime AWS-Zugriffsschlüssel, z. B. bPxRfi wjalrxUTNFEMI/K7MDENG/CYEXAMPLEKEY.

Git-Repository

Wenn Sie Git unter Source Control (Quellkontrolle) auswählen, werden die beiden folgenden zusätzlichen optionalen Einstellungen angezeigt:

Repository SSH key (Repository-SSH-Schlüssel)

Sie müssen einen SSH-Bereitstellungsschlüssel für den Zugriff auf private Git-Repositorys angeben. Bei Git-Submodulen muss der angegebene Schlüssel Zugriff auf diese Submodule haben. Weitere Informationen finden Sie unter Verwenden von Git-Repository-SSH-Schlüsseln.

> **Important**
>
> Für den SSH-Bereitstellungsschlüssel ist kein Passwort erforderlich. Stacks hat keine Möglichkeit, es weiterzugeben. AWS OpsWorks

Branch/Revision

Wenn das Repository mehrere Branches hat, lädt AWS OpsWorks Stacks standardmäßig den Master-Branch herunter. Um einen bestimmten Branch festzulegen, geben Sie den Branch-Namen (SHA1-Hash) oder den Tag-Namen ein. Um einen bestimmten Commit festzulegen, geben Sie die vollständige Commit-ID mit 40 Hexadezimalziffern an.

Subversion-Repository

Wenn Sie Subversion unter Source Control (Quellkontrolle) auswählen, werden die folgenden zusätzlichen Einstellungen angezeigt:

- Benutzername — Ihr Benutzername für private Repositorys.
- Passwort — Ihr Passwort für private Repositorys.
- Revision — [Optional] Der Revisionsname, falls Sie mehrere Revisionen haben.

  Um eine Verzweigung oder ein Tag anzugeben, müssen Sie die Repository-URL wie im folgenden Beispiel anpassen: **`http://repository_domain/repos/myapp/branches/my-apps-branch`** oder **`http://repository_domain_name/repos/calc/myapp/my-apps-tag`**.

Aktualisieren von benutzerdefinierten Rezeptbüchern

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn du AWS OpsWorks Stacks benutzerdefinierte Kochbücher zur Verfügung stellst, erstellen die integrierten Setup-Rezepte auf jeder neu gestarteten Instanz einen lokalen Cache und laden die Kochbücher in den Cache herunter. AWS OpsWorks Stacks führt dann Rezepte aus dem Cache aus, nicht aus dem Repository. Wenn Sie die benutzerdefinierten Kochbücher im Repository ändern, müssen Sie sicherstellen, dass die aktualisierten Kochbücher in den lokalen Caches Ihrer Instanzen installiert sind. AWS OpsWorks Stacks stellt automatisch die neuesten Kochbücher auf neuen Instanzen bereit, wenn diese gestartet werden. Für vorhandene Instances ist die Situation jedoch eine andere:

- Sie müssen aktualisierte benutzerdefinierte Rezeptbücher manuell auf Online-Instances bereitstellen.
- Sie müssen aktualisierte benutzerdefinierte Rezeptbücher nicht für Instance-Speicher-gestützte Offline-Instances bereitstellen, einschließlich last- und zeitbasierter Instances.

AWS OpsWorks Stacks stellt automatisch die aktuellen Kochbücher bereit, wenn die Instanzen neu gestartet werden.

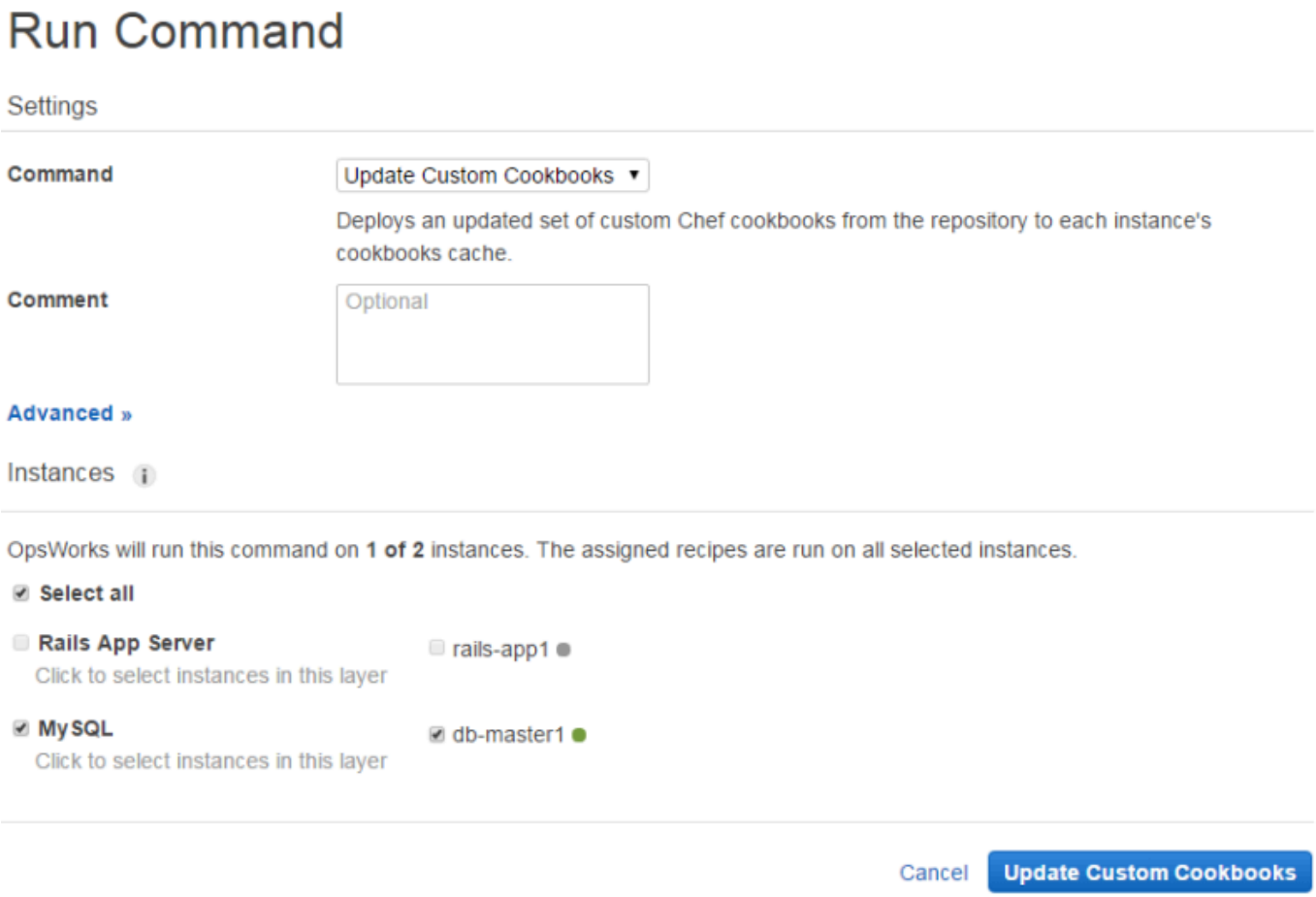
- Sie müssen EBS-gesicherte 24/7-Instances offline starten, die nicht last- oder zeitbasiert sind.
- Sie können Offline-EBS-gestützte last- und zeitbasierte Instances nicht starten, so dass es am einfachsten ist, Offline-Instances zu löschen und neue Instances hinzuzufügen, um diese zu ersetzen.

  Da es sich jetzt um neue Instanzen handelt, stellt AWS OpsWorks Stacks beim Start der Instanzen automatisch die aktuellen benutzerdefinierten Kochbücher bereit.

So aktualisieren Sie benutzerdefinierte Rezeptbücher:

1. Aktualisieren Sie Ihr Repository mit den geänderten Kochbüchern. AWS OpsWorks Stacks verwendet die Cache-URL, die Sie bei der ursprünglichen Installation der Kochbücher angegeben haben. Daher sollten sich der Name der Kochbuch-Stammdatei, der Speicherort des Repositorys und die Zugriffsrechte nicht ändern.

   - Ersetzen Sie bei Amazon S3- oder HTTP-Repositorys die ursprüngliche .zip-Datei durch eine neue .zip-Datei mit demselben Namen.
   - Für Git- oder Subversion-Repositorys, bearbeiten Sie Ihre Stack-Einstellungen, um das Feld Branch/Revision zur neuen Version zu ändern.
2. Klicken Sie auf der Seite des Stacks auf Run Command und wählen Sie den Befehl Update Custom Cookbooks aus.



3. Fügen Sie bei Bedarf einen Kommentar hinzu.
4. Geben Sie optional ein benutzerdefiniertes JSON-Objekt für den Befehl an, um der Stack-Konfiguration und den Bereitstellungsattributen, die AWS OpsWorks Stacks auf den Instances installiert, benutzerdefinierte Attribute hinzuzufügen. Weitere Informationen finden Sie unter Nutzen eines benutzerdefinierten JSON-Objekts und Überschreiben der Attribute.
5. Standardmäßig aktualisiert AWS OpsWorks Stacks die Kochbücher auf jeder Instanz. Um anzugeben, welche Instances zu aktualisieren ist, wählen Sie die entsprechenden Instances aus der Liste am Ende der Seite aus. Um alle Instances in einem Layer auszuwählen, wählen Sie das entsprechenden Layer-Kontrollkästchen in der linken Spalte aus.
6. Klicken Sie auf Benutzerdefinierte Kochbücher aktualisieren, um die aktualisierten Kochbücher zu installieren. AWS OpsWorks Stacks löscht die zwischengespeicherten benutzerdefinierten Kochbücher auf den angegebenen Instanzen und installiert die neuen Kochbücher aus dem Repository.

> **Note**
>
> Dieser Vorgang ist nur für vorhandene Instances erforderlich, die alte Versionen der Rezeptbücher in ihren Caches haben. Wenn Sie anschließend Instanzen zu einer Ebene hinzufügen, stellt AWS OpsWorks Stacks die Kochbücher bereit, die sich derzeit im Repository befinden, sodass sie automatisch die neueste Version erhalten.

Ausführen von Rezepten

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können Rezepte auf zwei Arten ausführen:

- Automatisch, indem Sie sie einem passenden Lebenszyklusereignis eines Layers zuweisen
- Manuell, indem Sie den Stack-Befehl "Execute Recipes" ausführen oder die Agent CLI verwenden

Themen

AWS OpsWorks Stapelt Lifecycle-Ereignisse

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Jeder Layer verfügt über fünf Lebenszyklusereignisse, denen jeweils Rezepte zugeordnet sind, die sich von Layer zu Layer unterscheiden. Wenn ein Ereignis auf einer Instance eines Layers auftritt, führt AWS OpsWorks Stacks die entsprechenden Rezepte automatisch aus. Implementieren Sie benutzerdefinierte Rezepte und weisen Sie sie den entsprechenden Ereignissen für jede Ebene zu, um auf diese Ereignisse individuell reagieren zu können. AWS OpsWorks Stacks führt diese Rezepte nach den integrierten Rezepten des Events aus.

Setup

Dieses Ereignis tritt nach dem Hochfahren einer Instance auf. Sie können das Setup Ereignis auch manuell auslösen, indem Sie den Befehl Setup stack verwenden. AWS OpsWorks Stacks führt Rezepte aus, die die Instanz entsprechend ihrer Ebene einrichten. Wenn die Instanz beispielsweise Mitglied der Rails App Server-Schicht ist, installieren die Setup Rezepte Apache, Ruby Enterprise Edition, Passenger und Ruby on Rails.

> **Note**
>
> Während eines Setup-Ereignisses muss eine Instance offline gehen. Da die Instance während des Setup-Lebenszyklusereignisses nicht den Status Online hat, werden Instances, auf denen Sie Setup-Ereignisse ausführen, vom Load Balancer getrennt.

Configure

Dieses Ereignis tritt auf allen Instances des Stacks auf, wenn eines der folgenden passiert:

- Eine Instance geht online oder offline.
- Sie ordnen einer Instance eine Elastic IP-Adresse zu oder Sie heben die Zuordnung einer Elastic IP-Adresse zu einer Instance auf.
- Sie fügen einem Layer einen Elastic Load Balancing Load Balancer hinzu oder trennen ihn von einem Layer.

Nehmen wir zum Beispiel an, Ihr Stack hat die Instanzen A, B und C und Sie starten eine neue Instance D. Nachdem D die Ausführung der Einrichtungsrezepte abgeschlossen hat, löst AWS OpsWorks Stacks das Configure Ereignis für A, B, C und D aus. Wenn Sie A anschließend

beenden, löst AWS OpsWorks Stacks das Configure Ereignis für B, C und D aus. AWS OpsWorks Stacks reagiert auf das Configure Ereignis, indem es die Configure Rezepte der einzelnen Ebenen ausführt, die die Konfiguration der Instanzen aktualisieren, sodass sie den aktuellen Satz von Online-Instances widerspiegelt. Das Configure-Ereignis ist daher ein guter Zeitpunkt, um Konfigurationsdateien wiederherzustellen. Beispielsweise konfigurieren die Configure HAProxy-Rezepte den Load Balancer neu, um Änderungen in der Gruppe der Online-Anwendungsserver-Instanzen zu berücksichtigen.

Sie können das Konfigurationsereignis auch manuell mithilfe des Stack-Befehls "Configure" auslösen.

Deploy

Dieses Ereignis tritt auf, wenn Sie den Befehl Deploy ausführen, um eine Anwendung für Anwendungsserver-Instances bereitzustellen. Auf den Instances werden Rezepte zur Bereitstellung der Anwendung und zugehöriger Dateien aus einem Repository für die Instances des Layers ausgeführt. Bei Rails-Anwendungsserver-Instances beispielsweise laden die Deploy-Rezepte eine bestimmte Ruby-Anwendung herunter und weisen Phusion Passenger an, diese neu zu laden. Sie können Deploy auch auf anderen Instances ausführen, um beispielsweise die Konfiguration der Instances zu aktualisieren und auf die neu bereitgestellte App abzustimmen.

> **Note**
>
> Der Befehl "Setup" beinhaltet den Befehl "Deploy", nach den Einrichtungsrezepten werden also auch die Bereitstellungsrezepte ausgeführt.

Undeploy

Dieses Ereignis tritt auf, wenn Sie eine App löschen oder den Befehl Undeploy ausführen, um eine Anwendung von Anwendungsserver-Instances zu löschen. Auf den angegebenen Instances werden Rezepte ausgeführt, um alle Anwendungsversionen zu löschen und die Instances zu bereinigen.

Shutdown

Dieses Ereignis tritt ein, nachdem Sie AWS OpsWorks Stacks angewiesen haben, eine Instance herunterzufahren, aber bevor die zugehörige Amazon EC2 EC2-Instance tatsächlich beendet wird. AWS OpsWorks Stacks führt Rezepte aus, um Bereinigungsaufgaben wie das Herunterfahren von Diensten auszuführen.

Wenn Sie dem Layer einen Elastic Load Balancing Load Balancer hinzugefügt und die Unterstützung für den Verbindungsabbau aktiviert haben, wartet AWS OpsWorks Stacks, bis der Verbindungsabbau abgeschlossen ist, bevor das Ereignis ausgelöst wird. Shutdown

Nach dem Auslösen eines Shutdown Ereignisses gewährt AWS OpsWorks Stacks den Shutdown Rezepten eine bestimmte Zeit, um ihre Aufgaben auszuführen, und stoppt oder beendet dann die Amazon EC2 EC2-Instance. Der Standardwert für den Shutdown-Timeout ist 120 Sekunden. Wenn Sie mehr Zeit benötigen, um Shutdown-Rezepte auszuführen, können Sie den Timeout-Wert in der Layer-Konfiguration anpassen. Weitere Informationen über Instance-Shutdown finden Sie unter Anhalten einer Instance.

> **Note**
>
> Ein Neustart einer Instance löst keine Lebenszyklusereignisse aus.

Weitere Erläuterungen zu den App-Befehlen Deploy und Undeploy finden Sie unter Bereitstellen von Anwendungen.

Nachdem eine Instance vollständig hochgefahren wurde, sieht die weitere Startup-Sequenz wie folgt aus:

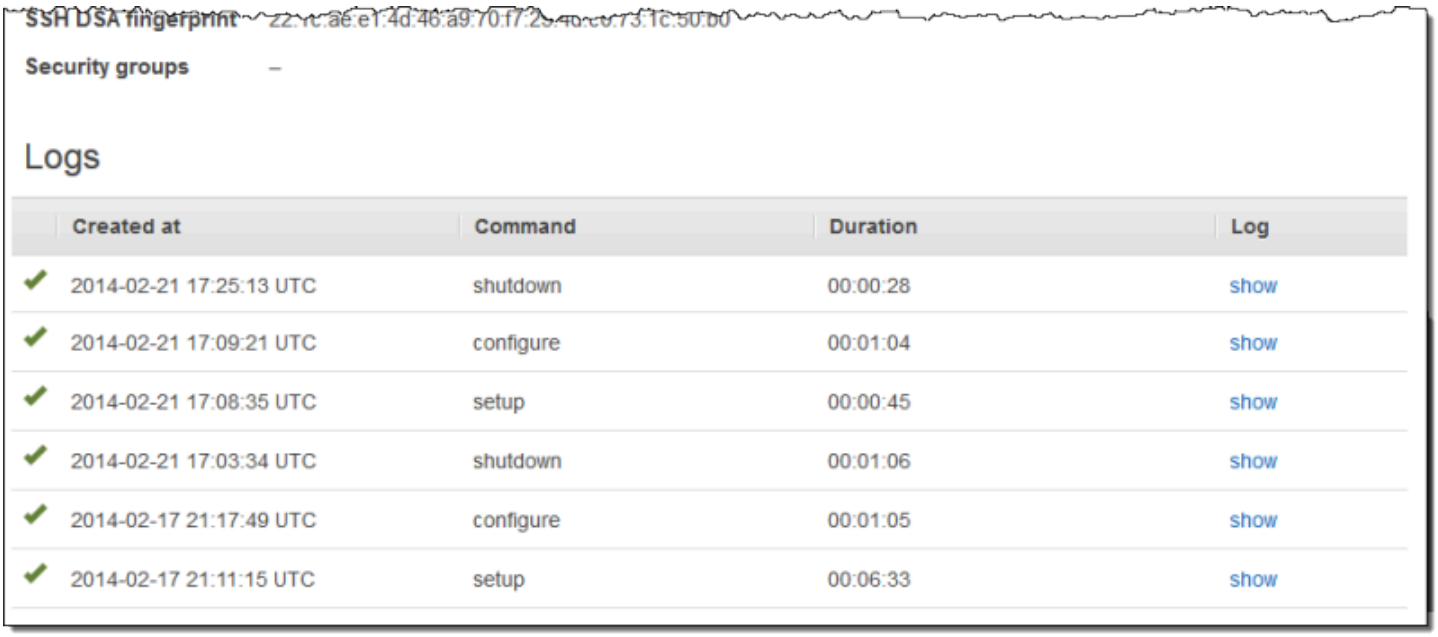
1. AWS OpsWorks Stacks führt die integrierten Setup Rezepte der Instance aus, gefolgt von allen benutzerdefinierten Rezepten. Setup
2. AWS OpsWorks Stacks führt die integrierten Deploy Rezepte der Instanz aus, gefolgt von allen benutzerdefinierten Deploy Rezepten.

   Die Instance ist jetzt online.
3. AWS OpsWorks Stacks löst ein Configure Ereignis auf allen Instanzen im Stack aus, einschließlich der neu gestarteten Instanz.

   AWS OpsWorks Stacks führt die integrierten Configure Rezepte der Instances aus, gefolgt von allen benutzerdefinierten Rezepten. Configure

> **Note**
>
> Um sich die Lebenszyklusereignisse anzusehen, die auf einer bestimmten Instance aufgetreten sind, rufen Sie die Seite Instances auf und klicken Sie auf den Namen der

Instance, um die Detailseite zu öffnen. Die Liste der Ereignisse finden Sie im Bereich Logs unten auf der Seite. Klicken Sie auf show in der Spalte Log, um das Chef-Protokoll für ein Ereignis anzusehen. Es enthält detaillierte Informationen zur Verarbeitung des Ereignisses einschließlich der ausgeführten Rezepte. Weitere Informationen zur Deutung der Chef-Protokolle finden Sie unter Chef-Protokolle.



Für jedes Lebenszyklusereignis installiert AWS OpsWorks Stacks auf jeder Instance eine Reihe von Stackkonfigurations- und Bereitstellungsattributen, die den aktuellen Stack-Status und bei Deploy Ereignissen Informationen zur Bereitstellung enthalten. Die Attribute enthalten außerdem auch Informationen zu den verfügbaren Instances, deren IP-Adressen usw. Weitere Informationen finden Sie unter Attribute für die Stack-Konfiguration und -Bereitstellung.

> **Note**
>
> Durch das gleichzeitige Starten oder Anhalten einer großen Anzahl von Instances kann es kurzfristig zu einer großen Anzahl von Configure-Ereignissen kommen. Um unnötige Verarbeitung zu vermeiden, reagiert AWS OpsWorks Stacks nur auf das letzte Ereignis. Die Stack-Konfigurations- und Bereitstellungsattribute des Ereignisses enthalten alle notwendigen Informationen zur Aktualisierung der Stack-Instances für alle anstehenden Änderungen. Dadurch entfällt die Notwendigkeit, auch die früheren Configure Ereignisse zu verarbeiten. AWS OpsWorks Stacks kennzeichnet die unverarbeiteten Configure Ereignisse als ersetzt.

Automatisches Ausführen von Rezepten

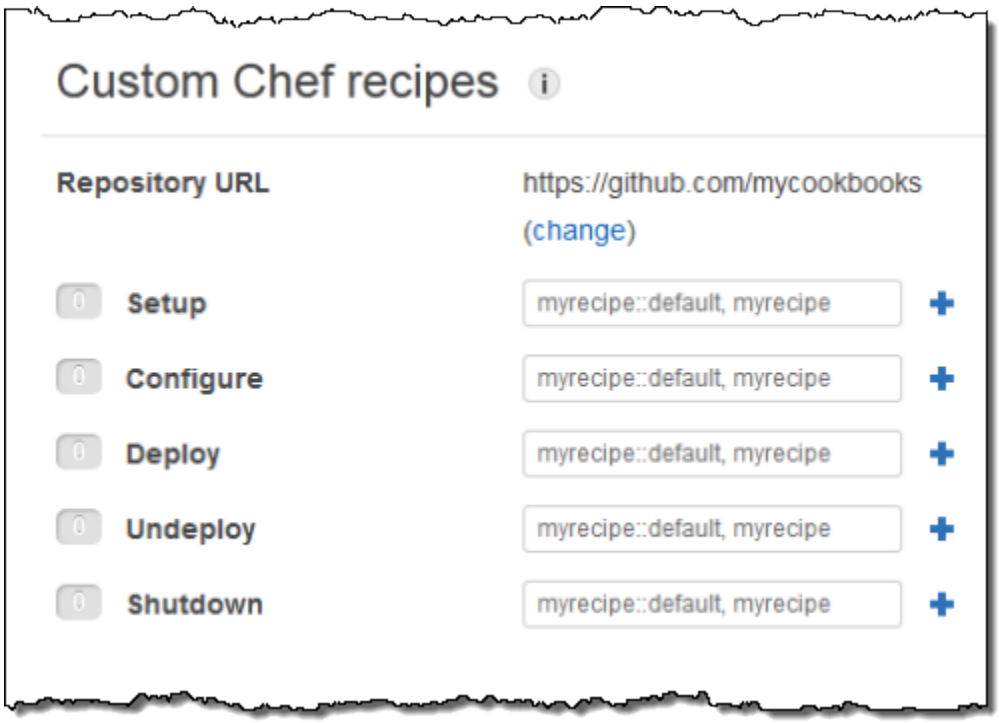
> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Jeder Layer verfügt über eine Reihe von integrierten Rezepten, die den einzelnen Lebenszyklusereignissen zugeordnet sind. Nicht alle Layer verfügen jedoch über Rezepte für "Bereitstellung aufheben". Wenn ein Lebenszyklusereignis auf einer Instance eintritt, führt AWS OpsWorks Stacks die entsprechenden Rezepte für die zugehörige Ebene aus.

Wenn Sie benutzerdefinierte Kochbücher installiert haben, können Sie AWS OpsWorks Stacks einige oder alle Rezepte automatisch ausführen lassen, indem Sie jedes Rezept dem Lebenszyklusereignis einer Ebene zuweisen. Nach dem Eintreten eines Ereignisses führt AWS OpsWorks Stacks die angegebenen benutzerdefinierten Rezepte nach den integrierten Rezepten der Ebene aus.

So weisen Sie benutzerdefinierte Rezepte zu den Ereignissen eines Layers hinzu

1. Klicken Sie auf der Seite Layers für den entsprechenden Layer auf Recipes und dann auf Edit. Wenn Sie noch keine benutzerdefinierten Rezeptbücher aktiviert haben, klicken Sie auf configure cookbooks, um die Seite Settings des Stacks zu öffnen. Wählen Sie für Use custom Chef Cookbooks Yes aus und geben Sie die Informationen zum Rezeptbuch-Repository ein. Klicken Sie nun auf Save und kehren Sie zur Bearbeitungsseite der Registerkarte Recipes zurück. Weitere Informationen finden Sie unter Installieren von benutzerdefinierten Rezeptbüchern.
2. Geben Sie auf der Registerkarte Recipes die benutzerdefinierten Rezepte in die entsprechenden Ereignisfelder ein und klicken Sie auf +, um sie zur Liste hinzuzufügen. Legen Sie Rezepte wie folgt fest: *Rezeptbuch*::*Rezept* (ohne die Erweiterung `.rb`).



Wenn Sie eine neue Instanz starten, führt AWS OpsWorks Stacks automatisch die benutzerdefinierten Rezepte für jedes Ereignis aus, nachdem die Standardrezepte ausgeführt wurden.

> **Note**
>
> Benutzerdefinierte Rezepte werden in der Reihenfolge ausgeführt, in der Sie auf der Konsole eingegeben wurden. Sie können auch ein Metarezept implementieren, um die Rezepte in einer bestimmten Reihenfolge auszuführen.

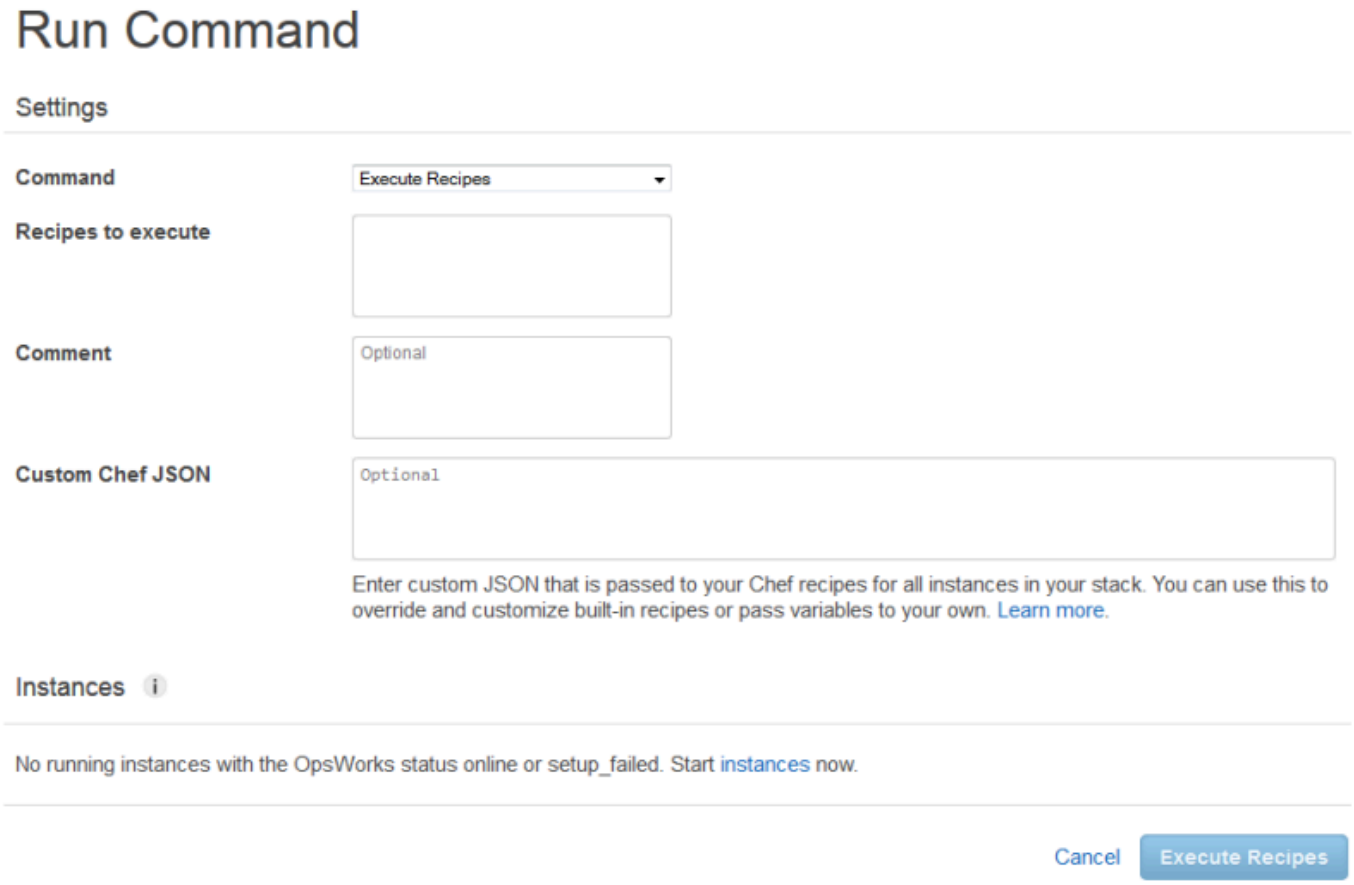
Manuelles Ausführen von Rezepten

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Rezepte werden zwar üblicherweise automatisch während eines Lebenszyklusereignisses ausgeführt, Sie können sie jedoch auch jederzeit manuell auf bestimmten oder allen Instances eines Stacks ausführen. Dies kann beispielsweise nützlich sein, um Aufgaben auszuführen, die sich keinem Lebenszyklusereignisse zuordnen lassen, wie eine Sicherung der Instances. Wenn Sie ein benutzerdefiniertes Rezept manuell ausführen möchten, muss es in einem Ihrer benutzerdefinierten Rezeptbücher enthalten, aber nicht unbedingt einem Lebenszyklusereignis zugeordnet sein. Wenn Sie ein Rezept manuell ausführen, installiert AWS OpsWorks Stacks dieselben `deploy` Attribute wie bei einem Deploy-Ereignis.

So führen Sie ein Rezept manuell auf Stack-Instances aus

1. Klicken Sie auf der Seite Stack auf Run command. Wählen Sie für Command die Option Execute Recipes aus.



2. Geben Sie im Feld Recipes to execute im Standardformat *`Rezeptbuchname`*::*`Rezeptname`* die Rezepte ein, die Sie ausführen möchten. Sie können mehrere Rezepte durch Kommas trennen. Die Rezepte werden in der eingegebenen Reihenfolge ausgeführt.
3. Fügen Sie optional im Feld Custom Chef JSON ein benutzerdefiniertes JSON-Objekt ein, um benutzerdefinierte Attribute festzulegen, die in die Stack-Konfigurations- und

Bereitstellungsattribute auf den Instances integriert werden. Weitere Informationen zur Verwendung von benutzerdefinierten JSON-Objekten finden Sie unter Nutzen eines benutzerdefinierten JSON-Objekts und Überschreiben der Attribute.

4. Wählen Sie unter Instances die Instanzen aus, auf denen AWS OpsWorks Stacks die Rezepte ausführen soll.

Wenn ein Lebenszyklusereignis eintritt, erhält der AWS OpsWorks Stacks-Agent einen Befehl zur Ausführung der zugehörigen Rezepte. Sie können diese Befehl auch manuell auf bestimmten Instances ausführen. Verwenden Sie dafür den entsprechenden Stack-Befehl oder den Befehl run_command der Agenten CLI. Weitere Informationen zur Verwendung der Agent CLI finden Sie unter AWS OpsWorks Stacks Agent CLI.

Ressourcenmanagement

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Auf der Seite Ressourcen können Sie die Elastic IP-Adresse, das Amazon EBS-Volume oder die Amazon RDS-Instance-Ressourcen Ihres Kontos in einem AWS OpsWorks Stacks-Stack verwenden. Über Resources (Ressourcen) können Sie die folgenden Aktionen ausführen:

- Registrieren einer Ressource mit einem Stack, damit Sie die Ressource einer der Stack-Instances zuweisen können.
- Zuweisen einer Ressource zu einer der Stack-Instances.
- Verschieben einer Ressource von einer Instance zu einer anderen.
- Trennen einer Ressource von einer Instance. Die Ressource bleibt registriert und kann einer anderen Instance zugewiesen werden.
- Abmelden einer Ressource. Eine nicht registrierte Ressource kann von AWS OpsWorks Stacks nicht verwendet werden, sie verbleibt jedoch in Ihrem Konto, sofern Sie sie nicht löschen, und kann bei einem anderen Stack registriert werden.

Bitte beachten Sie die folgenden Einschränkungen:

- Sie können registrierte Amazon EBS-Volumes nicht an Windows-Instances anhängen.
- Auf der Seite Ressourcen werden Standard-, PIOPS-, Throughput-Optimized HDD-, Cold HDD- oder General Purpose (SSD) Amazon EBS-Volumes verwaltet, jedoch keine RAID-Arrays.
- Amazon EBS-Volumes müssen xfs-formatiert sein.

  AWS OpsWorks Stacks unterstützt keine anderen Dateiformate wie ext4. Weitere Informationen zur Vorbereitung von Amazon EBS-Volumes finden Sie unter Bereitstellen eines Amazon EBS-Volumes zur Verwendung.
- Sie können ein Amazon EBS-Volume nicht an eine laufende Instance anhängen oder es von einer laufenden Instance trennen.

  Sie können nur mit Offline-Instances arbeiten. Sie können beispielsweise ein bereits verwendetes Volume mit einem Stack registrieren und einer Offline-Instance zuweisen, aber Sie müssen, bevor die neue Instance gestartet wird, die ursprüngliche Instance stoppen und das Volume von der Instance trennen. Andernfalls schlägt der Startprozess fehl.
- Alle registrierten Ressourcen werden ausschließlich in verwaltet. AWS OpsWorks Dadurch können Lebenszykluseigenschaften von Ressourcen außer Kraft gesetzt werden, z. B. `DeleteOnTermination` für EC2-Volumes.
- Sie können eine Elastic IP-Adresse einer bereits laufenden Instance nicht zuweisen oder von dieser trennen.

  Sie können mit Online- oder Offline-Instances arbeiten. Sie können beispielsweise eine verwendete Adresse registrieren und sie einer laufenden Instance zuweisen. AWS OpsWorks Stacks weist die Adresse dann automatisch neu zu.
- Zum Registrieren und Abmelden Ihrer Ressourcen muss Ihre IAM-Richtlinie Berechtigungen für die folgenden Aktionen erteilen:

Amazon EBS-Volumes	Elastic IP-Adressen	Amazon RDS-Instances
RegisterVolume	RegisterElasticIp	RegisterRdsDbInstance
UpdateVolume	UpdateElasticIp	UpdateRdsDbInstance
DeregisterVolume	DeregisterElasticIp	DeregisterRdsDbInstance

Die Ebene zum Verwalten von Berechtigungen erteilt Berechtigungen für alle folgenden Aktionen. Um zu verhindern, dass ein verwalteter Benutzer bestimmte Ressourcen registriert oder abmeldet, bearbeiten Sie die IAM-Richtlinie so, dass sie für die entsprechenden Aktionen keine Berechtigungen erteilt. Weitere Informationen finden Sie unter Sicherheit und Berechtigungen.

Themen

Registrieren von Ressourcen mit einem Stack

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Amazon EBS-Volumes oder Elastic IP-Adressen müssen bei einem Stack registriert werden, bevor Sie sie an Instances anhängen können. Wenn AWS OpsWorks Stacks Ressourcen für einen Stack erstellt, werden sie automatisch bei diesem Stack registriert. Wenn Sie extern erstellte Ressourcen verwenden möchten, müssen Sie diese explizit registrieren. Beachten Sie Folgendes:

- Sie können eine Ressource immer nur jeweils mit einem Stack registrieren.
- Wenn Sie einen Stack löschen, hebt AWS OpsWorks Stacks die Registrierung aller Ressourcen auf.

Themen

- Registrieren von Elastic IP-Adressen mit einem Stack
- Registrierung von Amazon RDS-Instances mit einem Stack

Registrierung von Amazon EBS-Volumes mit einem Stack

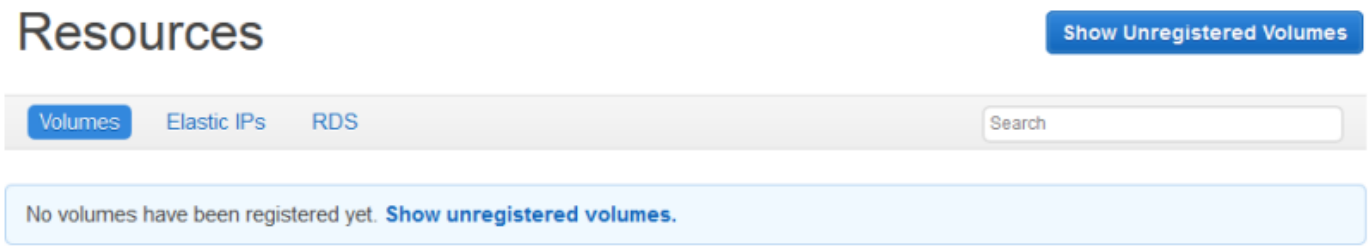
> **Note**
>
> Diese Ressource kann nur mit Linux-Stacks verwendet werden. Sie können ein Amazon EBS-Volume zwar bei einem Windows-Stack registrieren, aber Sie können es nicht an eine Instance anhängen.

Sie können die Seite Ressourcen verwenden, um ein Amazon EBS-Volume bei einem Stack zu registrieren. Dabei gelten die folgenden Einschränkungen:

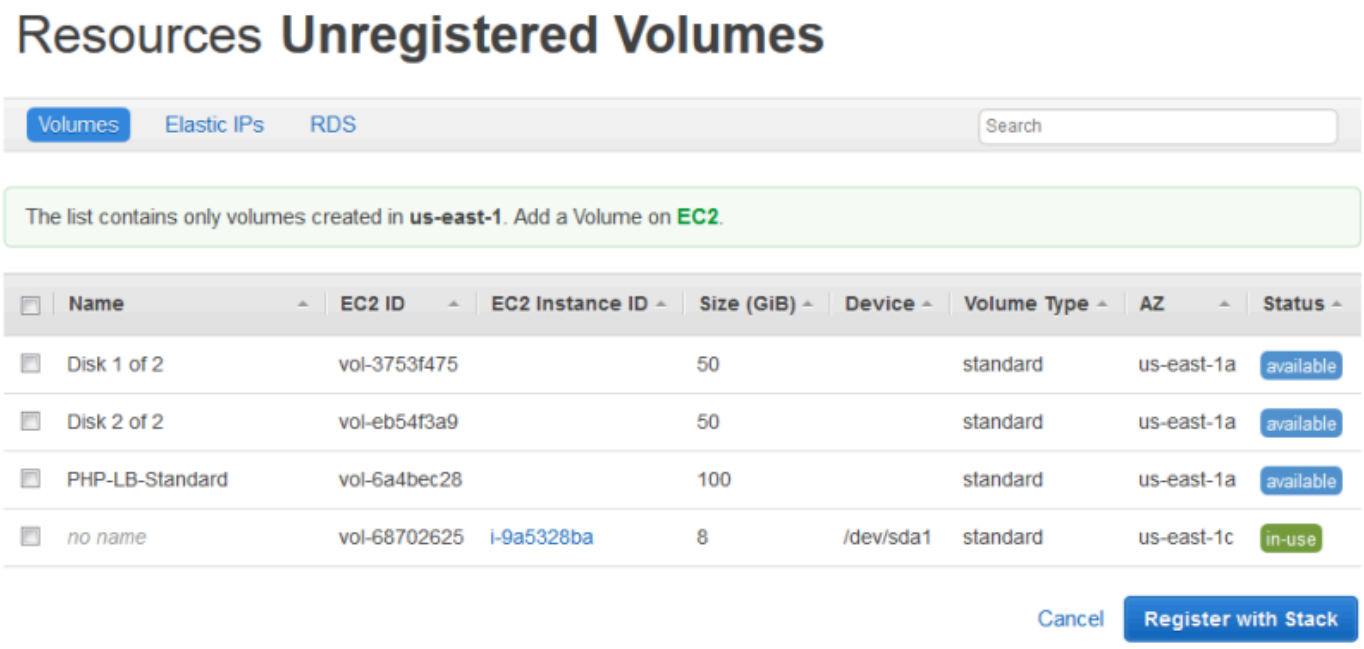
- Angehängte Amazon EBS-Volumes ohne Root-Rechte müssen standardmäßige, durchsatzoptimierte HDD, Cold HDD, PIOPS oder General Purpose (SSD) sein, aber kein RAID-Array. Informationen den maximalen und Mindestgrößen von Volumes finden Sie unter EBS-Datenträger in diesem Handbuch.
- Volumes müssen XFS-formatiert sein..
- AWS OpsWorks Stacks unterstützt keine anderen Dateiformate, wie z. B. Fourth Extended File System (ext4), für Amazon EBS-Volumes, die keine Root-Volumes sind. Weitere Informationen zur Vorbereitung von Amazon EBS-Volumes finden Sie unter Ein Amazon EBS-Volume zur Verwendung verfügbar machen. Beachten Sie, dass das in diesem Abschnitt gezeigte Beispiel die Erstellung eines ext4-basiertem Volumes beschreibt. Sie können die gleichen Schritte jedoch auch für XFS-basierte Volumes ausführen.

Um ein Amazon EBS-Volume zu registrieren

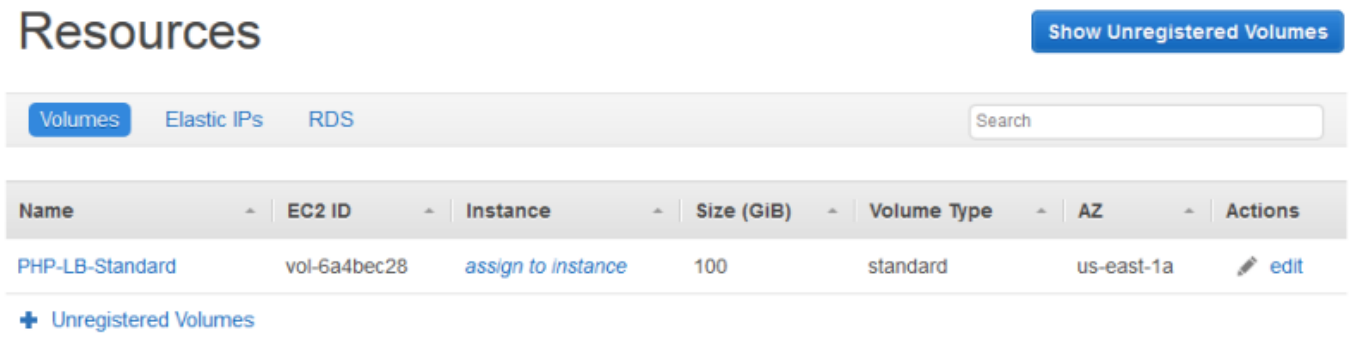
1. Öffnen Sie den gewünschten Stack und klicken Sie im Navigationsbereich auf Resources (Ressourcen).
2. Klicken Sie auf Volumes, um die verfügbaren Amazon EBS-Volumes anzuzeigen. Zu Beginn hat der Stack keine registrierten Volumes, wie in der folgenden Abbildung dargestellt ist.



3. Klicken Sie auf Nicht registrierte Volumes anzeigen, um die Amazon EBS-Volumes in Ihrem Konto anzuzeigen, die sich in der Region des Stacks befinden, und gegebenenfalls die VPC des Stacks. Die Spalte Status gibt an, ob die Volumes verwendungsbereit sind. Volume Type (Volume-Typ) gibt an, ob es sich um ein Standard-(`standard`), Allzweck SSD-(`gp2`), PIOPS-(`io1`, gefolgt von der Angabe für IOPS pro Festplatte in Klammern), Throughput Optimized HDD-(`st1`) oder Cold HDD-Volume (`sc1`) handelt.



4. Wählen Sie die entsprechenden Volumes aus und klicken Sie auf Register to Stack (Für Stack registrieren). Die Seite Resources (Ressourcen) führt jetzt die neu registrierten Volumes auf.



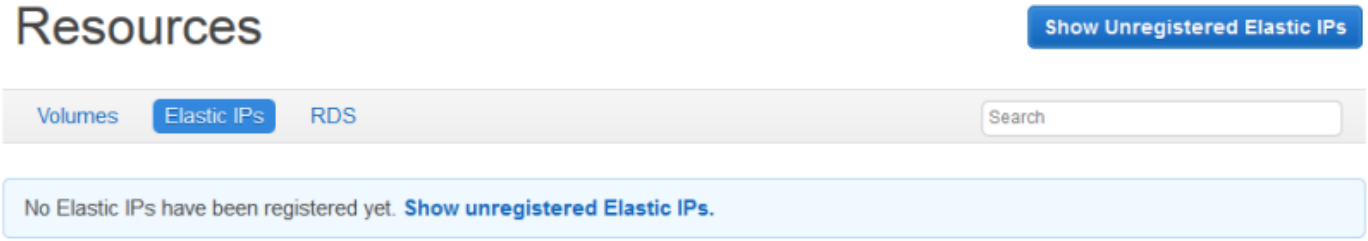
Wenn Sie weitere Volumes registrieren möchten, klicken Sie auf Show Unregistered Volumes (Unregistrierte Volumes anzeigen) oder + Unregistered Volumes (+ unregistrierte Volumes) und wiederholen Sie den Vorgang.

Registrieren von Elastic IP-Adressen mit einem Stack

Führen Sie die folgenden Schritte aus, um Elastic IP-Adressen zu registrieren.

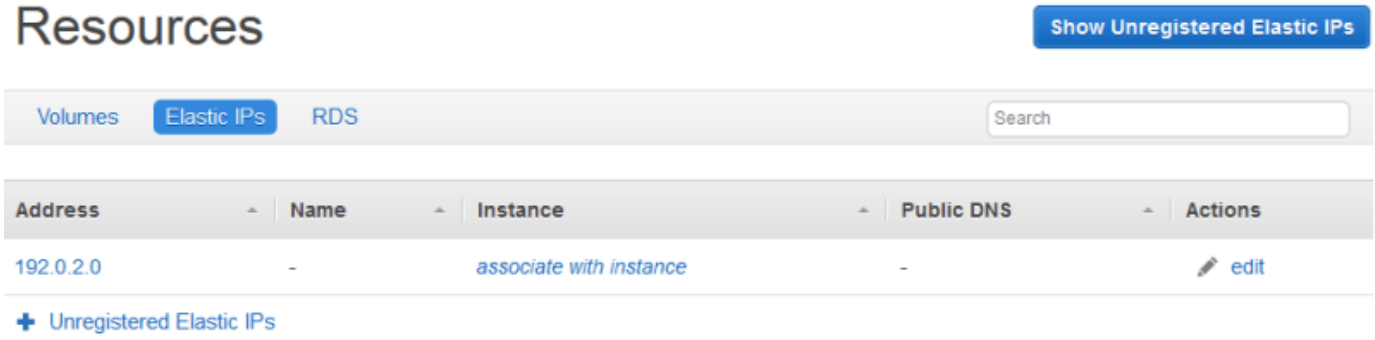
Registrieren einer Elastic IP-Adresse

1. Öffnen Sie die Seite Resources (Ressourcen) des Stacks und klicken Sie auf Elastic IPs (Elastic IP-Adressen), um die verfügbaren Elastic IP-Adressen anzuzeigen. Zu Beginn hat der Stack keine registrierten Adressen, wie in der folgenden Abbildung dargestellt ist.



2. Klicken Sie auf Show Unregistered Elastic IPs (Unregistrierte Elastic IP-Adressen anzeigen), um die verfügbaren Elastic IP-Adressen in Ihrem Konto anzuzeigen, die sich in der Stack-Region befinden.

3. Wählen Sie die entsprechenden Adressen aus und klicken Sie auf Register to Stack (Für Stack registrieren). Damit werden Sie zurückgeführt auf die Seite Resources (Ressourcen), wo jetzt die neu registrierten Adressen aufgeführt sind.



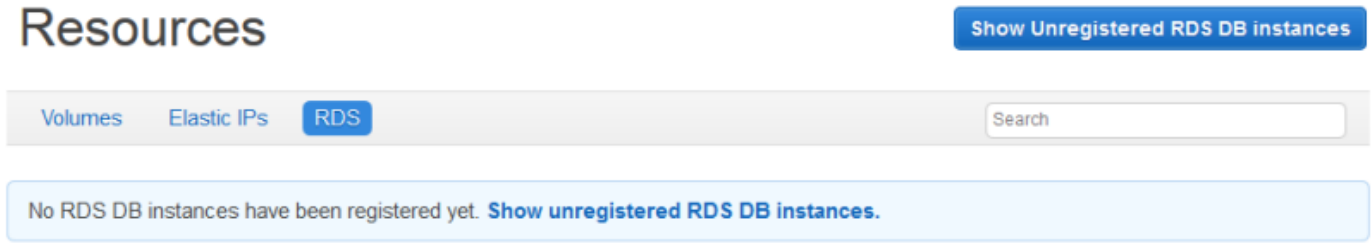
Wenn Sie weitere Adressen registrieren möchten, klicken Sie auf Show Unregistered Elastic IPs (Unregistrierte Elastic IP-Adressen anzeigen) oder + Unregistered Elastic IPs (+ unregistrierte Elastic IP-Adressen) und wiederholen Sie den Vorgang.

Registrierung von Amazon RDS-Instances mit einem Stack

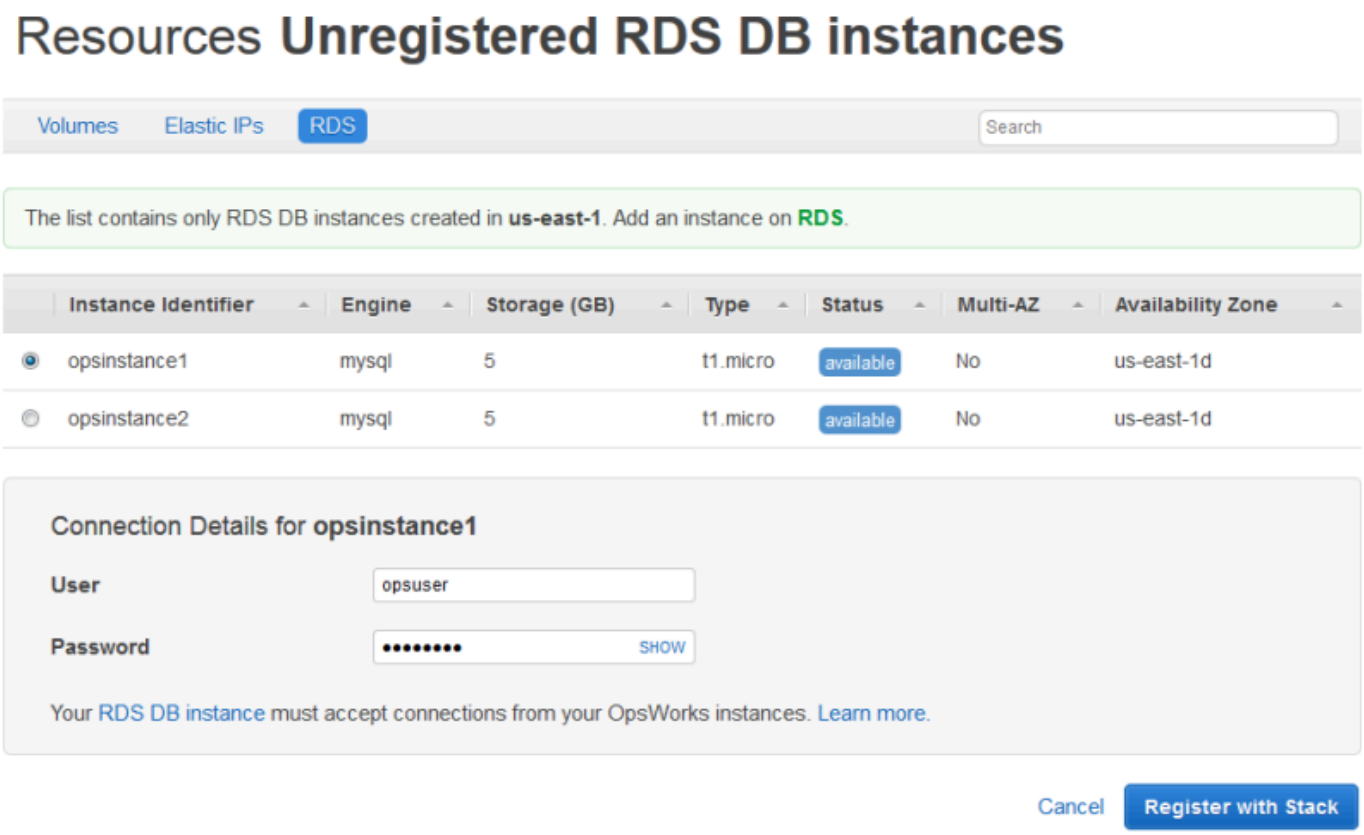
Gehen Sie wie folgt vor, um Amazon RDS-Instances zu registrieren.

Um eine Amazon RDS-Instance zu registrieren

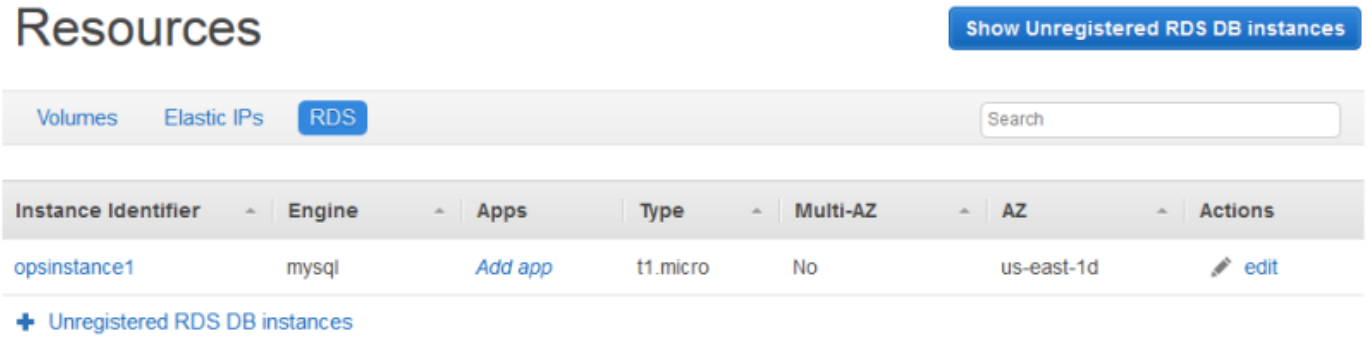
1. Öffnen Sie die Ressourcenseite des Stacks und klicken Sie auf RDS, um die verfügbaren Amazon RDS-Instances anzuzeigen. Zu Beginn hat der Stack keine registrierten Instances, wie in der folgenden Abbildung dargestellt ist.



2. Klicken Sie auf Nicht registrierte RDS-DB-Instances anzeigen, um die verfügbaren Amazon RDS-Instances in Ihrem Konto anzuzeigen, die sich in der Region des Stacks befinden.



3. Wählen Sie die geeignete Instance aus, geben Sie den Master-Benutzer und das Master-Passwort bei User (Benutzer) und Password (Passwort) ein und klicken Sie auf Register to Stack (Für Stack registrieren). Damit werden Sie zurückgeführt auf die Seite Resources (Ressourcen), wo jetzt die neu registrierte Instance aufgeführt ist.



> **Important**
>
> Sie müssen sicherstellen, dass der Benutzer und das Passwort, die Sie zur Registrierung der Amazon RDS-Instance verwenden, einem gültigen Benutzer und Passwort

entsprechen. Ist dies nicht der Fall, können die Anwendungen keine Verbindung zur Instance herstellen.

Wenn Sie weitere Adressen registrieren möchten, klicken Sie auf Show Unregistered RDS DB instances (Unregistrierte RDS DB-Instances anzeigen) oder + Unregistered RDS DB instances (+ unregistrierte RDS DB-Instances) und wiederholen Sie den Vorgang. Weitere Informationen zur Verwendung von Amazon RDS-Instances mit AWS OpsWorks Stacks finden Sie unterAmazon RDS-Serviceschicht.

Note

Sie können Amazon RDS-Instances auch über die Seite Layers registrieren. Weitere Informationen finden Sie unter Amazon RDS-Serviceschicht.

Zuweisen und Verschieben von Ressourcen

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Nachdem Sie eine Ressource mit einem Stack registriert haben, können Sie diese einer der Stack-Instances zuweisen. Sie können eine zugewiesene Ressource auch von einer Instance zu einer anderen verschieben. Beachten Sie Folgendes:

- Wenn Sie Amazon EBS-Volumes anhängen oder verschieben, müssen die an dem Vorgang beteiligten Instances offline sein. Wenn sich die gewünschte Instance nicht auf der Seite Resources (Ressourcen) befindet, rufen Sie die Seite Instances auf und stoppen Sie die Instance. Nachdem die Instance gestoppt wurde, können Sie zur Seite Resources (Ressourcen) zurückkehren und die Ressource zuweisen oder verschieben.

- Wenn Sie Elastic IP-Adressen zuweisen oder verschieben, können die Instances online oder offline sein.
- Wenn Sie eine Instance löschen, bleiben alle zugewiesenen Ressourcen mit dem Stack registriert. Sie können dann die Ressource einer anderen Instanz zuweisen oder die Ressource, wenn Sie sie nicht mehr benötigen, abmelden.

Themen

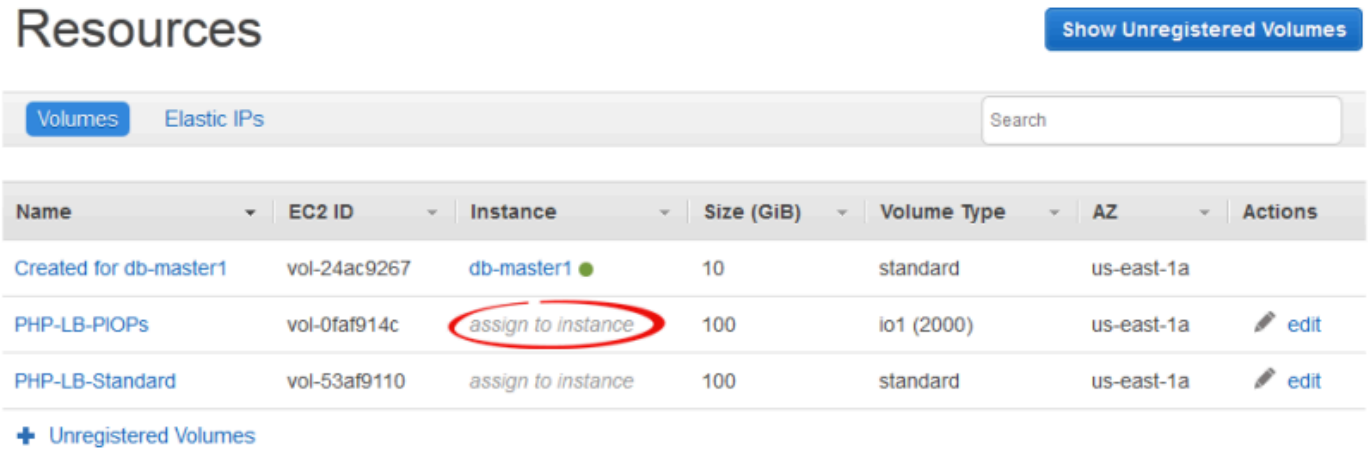
Zuweisen von Amazon EBS-Volumes zu einer Instance

> **Note**
>
> Sie können Windows-Instances keine Amazon EBS-Volumes zuweisen.

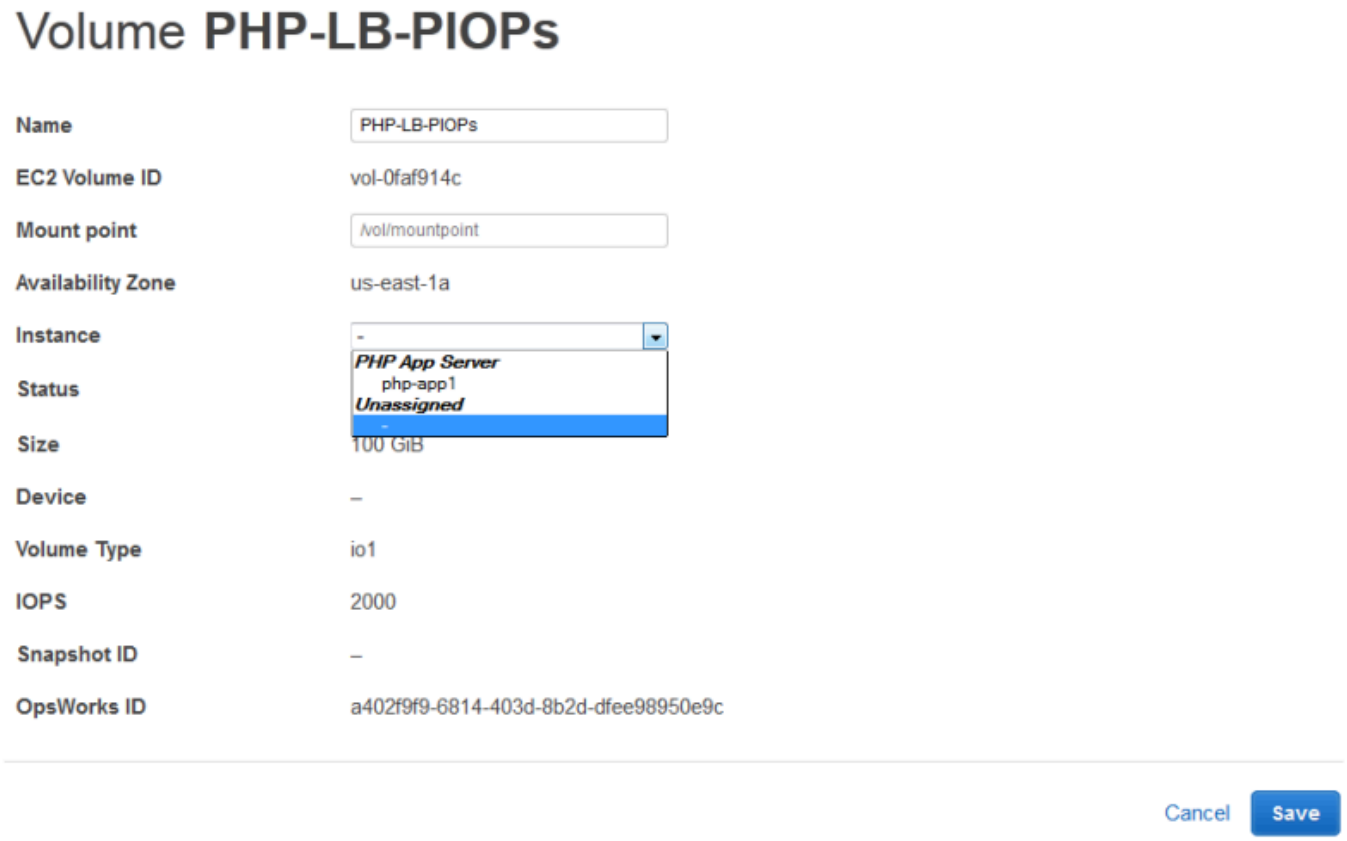
Sie können einer Instance ein registriertes Amazon EBS-Volume zuweisen und es von einer Instance auf eine andere verschieben, aber beide Instances müssen offline sein.

So weisen Sie einer Instance ein Amazon EBS-Volume zu

1. Klicken Sie auf der Seite "Ressourcen" auf assign to instance (der Instance zuweisen) in der Spalte Instance des zutreffenden Volumes.



2. Wählen Sie auf der Seite "Volume-Details" die entsprechende Instance aus, geben Sie den Namen und den Mounting-Punkt des Volumes an und klicken Sie auf Save (Speichern), um der Instance das Volume zuzuweisen.



> **Important**
>
> Wenn Sie Ihrer Instance ein externes, in Gebrauch befindliches Volume zugewiesen haben, müssen Sie die Amazon EC2 EC2-Konsole, API oder CLI verwenden, um die Zuweisung zur ursprünglichen Instance aufzuheben. Andernfalls schlägt der Startvorgang fehl.

Sie können die Detailseite auch verwenden, um ein zugewiesenes Amazon EBS-Volume auf eine andere Instance im Stack zu verschieben.

Um ein Amazon EBS-Volume auf eine andere Instance zu verschieben

1. Stellen Sie sicher, dass beide Instances offline sind.

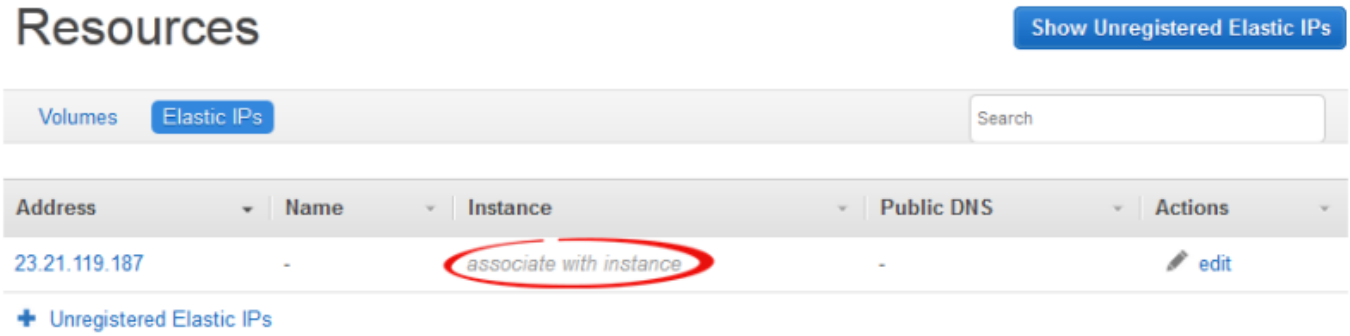
2. Klicken Sie auf der Seite Resources (Ressourcen) auf Volumes und dann auf edit (Bearbeiten) in der Spalte Actions (Aktionen) des Volumes.
3. Führen Sie eine der folgenden Aktionen aus:
   - Wenn Sie das Volume zu einer anderen Instance im Stack verschieben wollen, wählen Sie die entsprechende Instance aus der Liste Instance aus und klicken Sie auf Save (Speichern).
   - Wenn Sie das Volume zu einer Instance in einem anderen Stack verschieben wollen, melden Sie das Volume ab, registrieren Sie das Volume mit dem neuen Stack und weisen Sie es der neuen Instance zu.

Zuordnen von Elastic IP-Adressen zu einer Instance

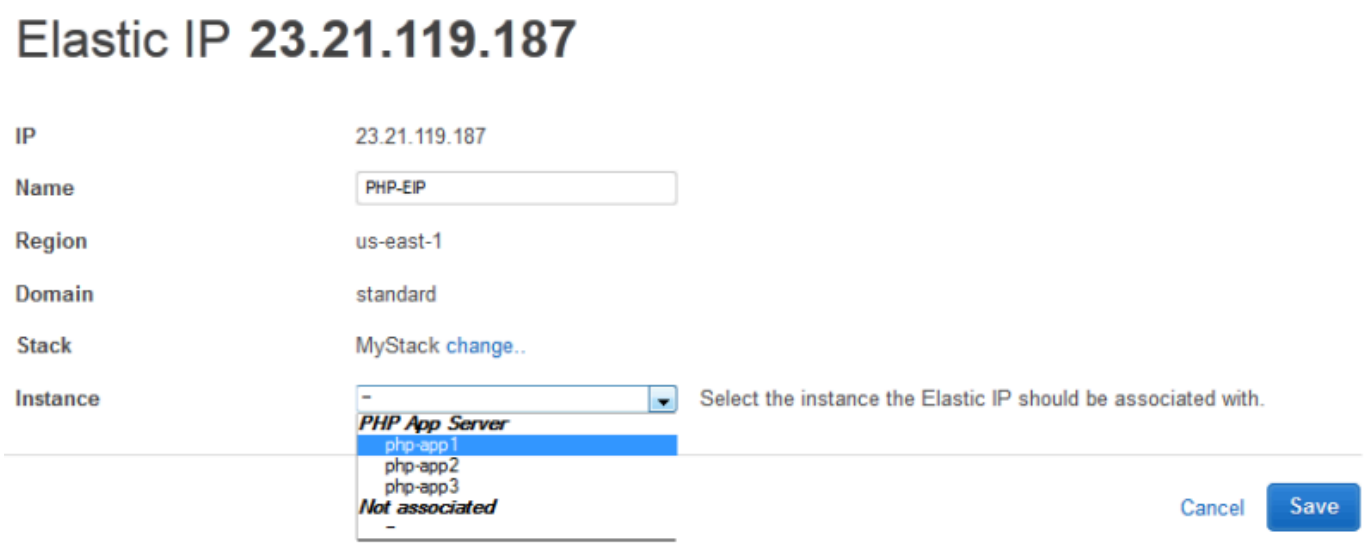
Sie können eine registrierte Elastic IP-Adresse einer Instance zuordnen und sie von einer Instance zu einer anderen zu verschieben, einschließlich Instances in anderen Stacks. Die Instances können entweder online oder offline sein.

Zuordnen einer Elastic IP-Adresse zu einer Instance

1. Klicken Sie auf der Seite Resources (Ressourcen) auf associate with instance (der Instance zuordnen) in der Spalte Instance der zutreffenden Adresse.



2. Klicken Sie auf der Detailseite der Adresse auf die entsprechende Instance, geben Sie den Namen der Adresse an und klicken Sie auf Save (Speichern), um der Instance die Adresse zuzuordnen.



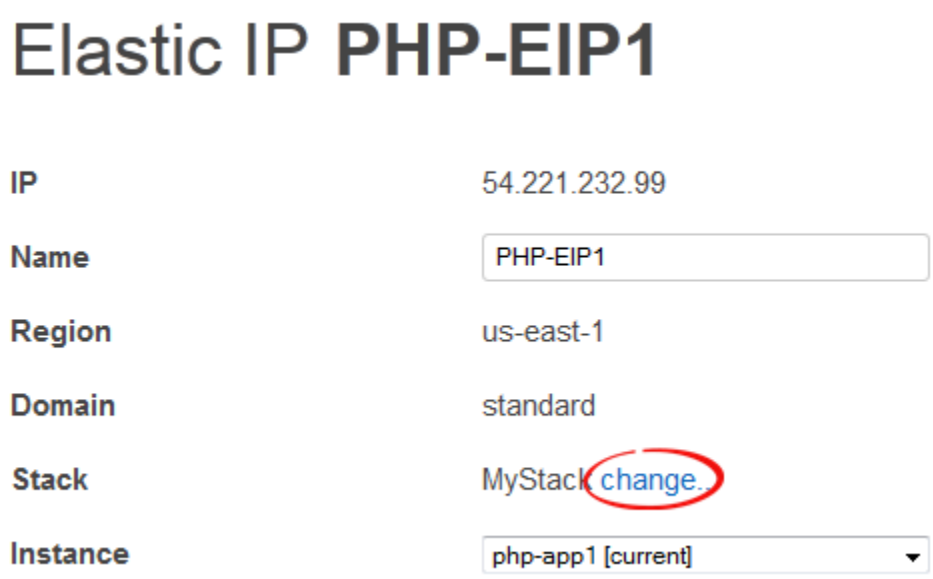
> **Note**
>
> Wenn die Elastic IP-Adresse derzeit mit einer anderen Online-Instance verknüpft ist, weist AWS OpsWorks Stacks die Adresse automatisch der neuen Instance neu zu.

Sie können die Seite "Details" auch verwenden, um eine zugeordnete Elastic IP-Adresse zu einer anderen Instance zu verschieben.

Verschieben einer Elastic IP-Adresse zu einer anderen Instance

1. Klicken Sie auf der Seite Resources (Ressourcen) auf Elastic IPs (Elastic IP-Adressen) und dann auf edit (Bearbeiten) in der Spalte Actions (Aktionen) der Adresse.
2. Führen Sie eine der folgenden Aktionen aus:

   - Wenn Sie die Adresse zu einer anderen Instance im Stack verschieben wollen, wählen Sie die entsprechende Instance aus der Liste Instance aus und klicken Sie auf Save (Speichern).
   - Um die Adresse auf eine Instance in einem anderen Stack zu verschieben, klicken Sie in den Stack-Einstellungen auf Ändern, um eine Liste der verfügbaren Stacks zu sehen. Wählen Sie einen Stack aus der Liste Stack und eine Instance aus der Liste Instance aus. Klicken Sie dann auf Save (Speichern).



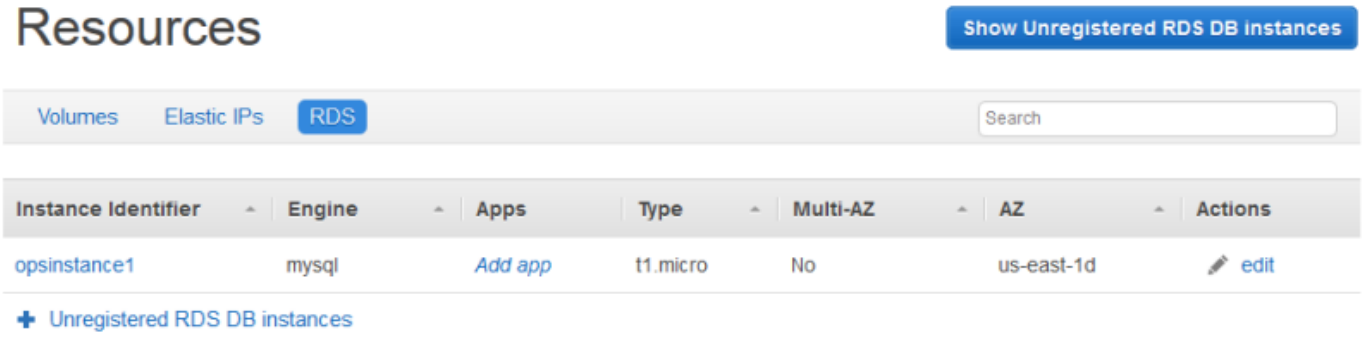
Nachdem Sie eine Adresse angehängt oder verschoben haben, löst AWS OpsWorks Stacks ein Configure-Lifecycle-Ereignis aus, um die Instanzen des Stacks über die Änderung zu informieren.

Amazon RDS-Instances an eine App anhängen

Sie können eine Amazon RDS-Instance an eine oder mehrere Apps anhängen.

So hängen Sie eine Amazon RDS-Instance an eine App an

1. Klicken Sie auf der Seite Resources (Ressourcen) auf Add app (App hinzufügen) in der Spalte Apps der zutreffenden Instance.



2. Verwenden Sie die Seite „App hinzufügen“, um die Amazon RDS-Instance anzuhängen. Weitere Informationen finden Sie unter Hinzufügen von Apps.

Da ein Amazon RDS an mehrere Apps angehängt werden kann, gibt es kein spezielles Verfahren, um die Instance von einer App in eine andere zu verschieben. Sie können also entweder die erste Anwendung bearbeiten, um die RDS-Instance zu entfernen, oder die zweite Anwendung, um die RDS-Instance hinzuzufügen. Weitere Informationen finden Sie unter Bearbeiten von Anwendungen.

Trennen von Ressourcen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn Sie eine zugewiesene Ressource nicht mehr benötigen, können Sie sie trennen. Diese Ressource bleibt mit dem Stack registriert und kann an anderer Stelle zugewiesen werden.

Themen

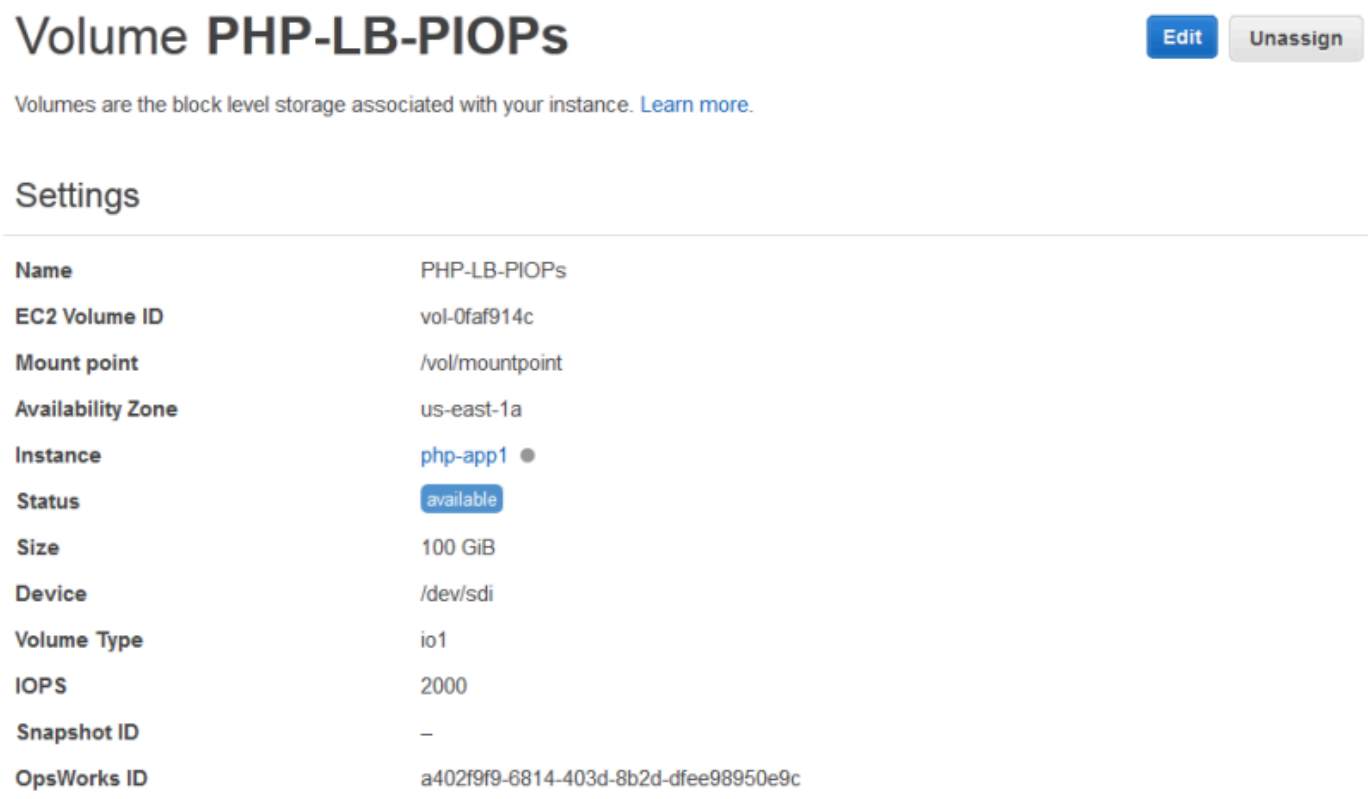
- Aufheben der Zuweisung von Amazon EBS-Volumes
- Aufheben von Zuordnungen von Elastic IP-Adressen
- Trennen von Amazon RDS-Instances

Aufheben der Zuweisung von Amazon EBS-Volumes

Gehen Sie wie folgt vor, um die Zuweisung eines Amazon EBS-Volumes zu seiner Instance aufzuheben.

So heben Sie die Zuweisung eines Amazon EBS-Volumes auf

1. Stellen Sie sicher, dass die Instance offline ist.
2. Klicken Sie auf der Seite Resources (Ressourcen) auf Volumes und klicken Sie auf den Namen des Volumes.
3. Klicken Sie auf der Detailseite des Volumes auf Unassign (Zuweisung aufheben).

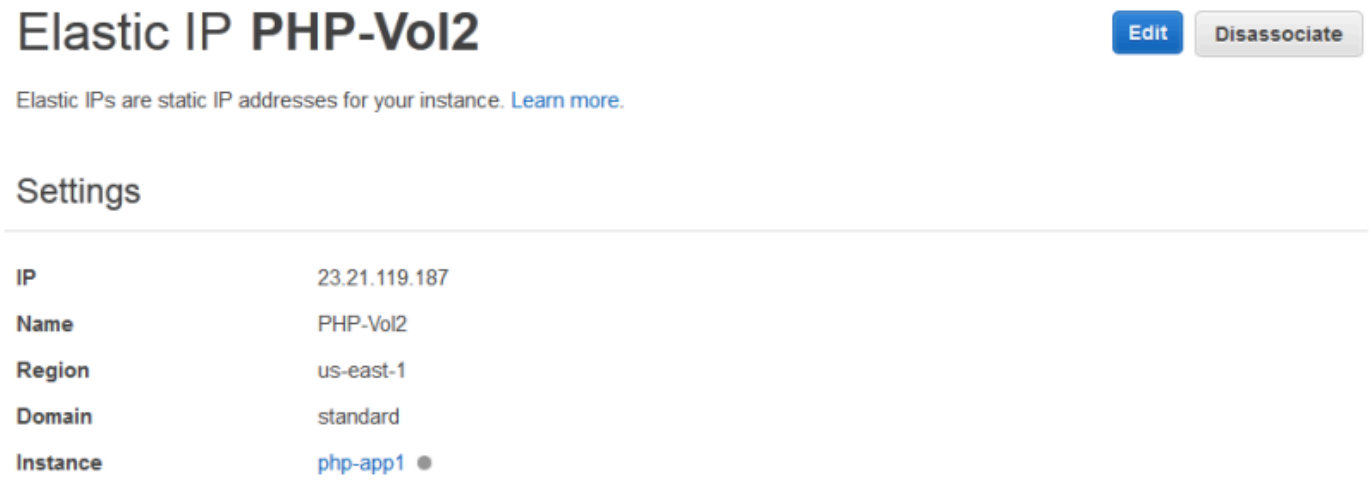


Aufheben von Zuordnungen von Elastic IP-Adressen

Führen Sie die folgenden Schritte aus, um die Zuordnung einer Elastic IP-Adresse zu einer Instance aufzuheben.

So heben Sie die Zuordnung einer Elastic-IP-Adresse auf

1. Klicken Sie auf der Seite Resources (Ressourcen) auf Elastic IPs (Elastic IP-Adressen) und dann auf edit (Bearbeiten) in der Spalte Actions (Aktionen) der Adresse.
2. Klicken Sie auf der Detailseite der Adresse auf Disassociate (Zuordnen aufheben).



Nachdem Sie die Zuordnung einer Adresse aufgehoben haben, löst AWS OpsWorks Stacks ein Configure-Lifecycle-Ereignis aus, um die Instances des Stacks über die Änderung zu informieren.

Trennen von Amazon RDS-Instances

Gehen Sie wie folgt vor, um einen Amazon RDS von einer App zu trennen.

So trennen Sie eine Amazon RDS-Instance

1. Klicken Sie auf der Seite Resources (Ressourcen) auf RDS und dann auf die entsprechende Anwendung in der Spalte Apps.
2. Klicken Sie auf Edit (Bearbeiten) und bearbeiten Sie die Anwendungskonfiguration, um die Instance zu trennen. Weitere Informationen finden Sie unter Bearbeiten von Anwendungen.

> **Note**
>
> Mit diesem Verfahren wird ein Amazon RDS von einer einzelnen App getrennt. Wenn die Instance mehreren Anwendungen zugewiesen ist, müssen Sie diese Vorgehensweise für jede Anwendung wiederholen.

Abmelden von Ressourcen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn es nicht mehr nötig ist, dass eine Ressource mit einem Stack registriert ist, können Sie diese abmelden. Durch die Abmeldung wird die Ressource nicht aus deinem Konto gelöscht. Sie verbleibt dort und kann bei einem anderen Stack registriert oder außerhalb von Stacks verwendet werden. AWS OpsWorks Wenn Sie die Ressource vollständig löschen möchten, haben Sie zwei Möglichkeiten:

- Wenn eine Elastic IP- oder Amazon EBS-Ressource an eine Instance angehängt ist, können Sie die Ressource löschen, wenn Sie die Instance löschen.

  Rufen Sie die Seite Instances auf, klicken Sie auf delete (Löschen) in der Spalte Actions (Aktionen) der Instance und wählen Sie dann Delete instance's EBS volumes (EBS-Volumes der Instance löschen) oder Delete the instance's Elastic IP (Elastic IP-Adresse der Instance löschen) aus.
- Melden Sie die Ressource ab und verwenden Sie dann die Amazon EC2- oder Amazon RDS-Konsole, API oder CLI, um sie zu löschen.

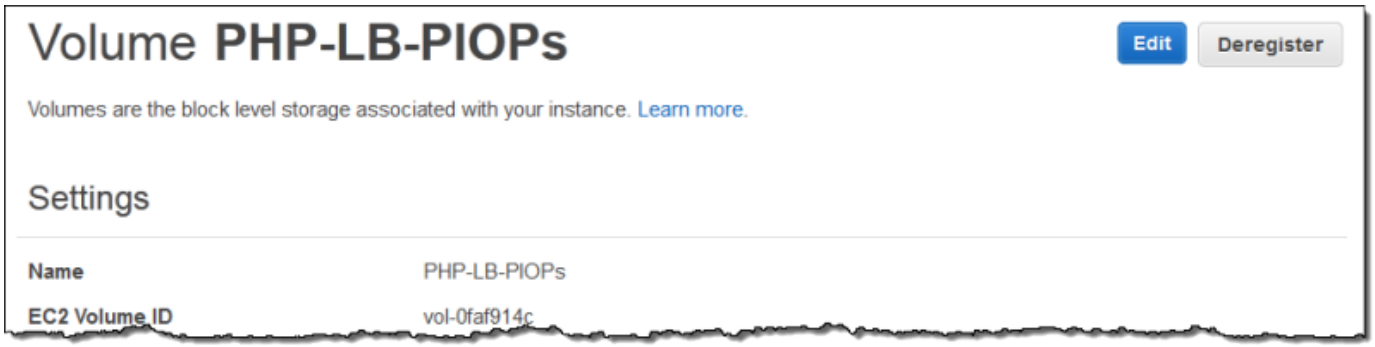
Themen

Abmeldung von Amazon EBS-Volumes

Gehen Sie wie folgt vor, um ein Amazon EBS-Volume abzumelden.

So melden Sie ein Amazon EBS-Volume ab

1. Wenn das Volume einer Instance zugewiesen wurde, heben Sie die Zuweisung auf, wie in Aufheben der Zuweisung von Amazon EBS-Volumes beschrieben.
2. Klicken Sie auf der Seite Resources (Ressourcen) auf den Namen des Volumes in der Spalte Name.
3. Klicken Sie auf der Detailseite des Volumes auf Deregister (Abmelden).

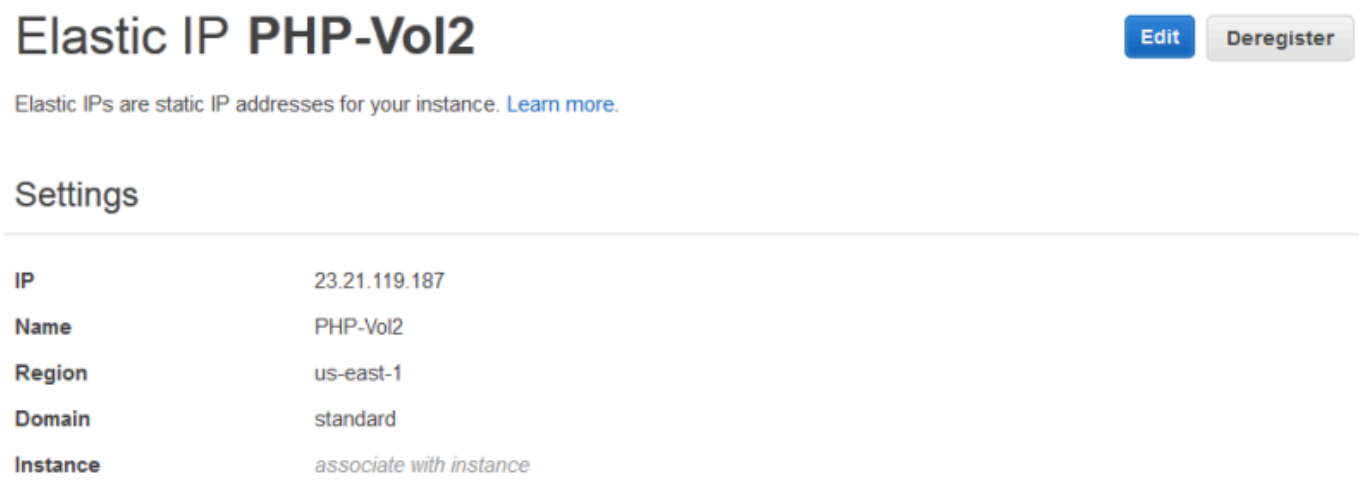


Abmelden von Elastic IP-Adressen

Führen Sie die folgenden Schritte aus, um eine Elastic IP-Adresse abzumelden.

Abmelden einer Elastic IP-Adresse

1. Wenn die Adresse einer Instance zugeordnet ist, heben Sie die Zuordnung auf, wie in Aufheben von Zuordnungen von Elastic IP-Adressen beschrieben.
2. Klicken Sie auf der Seite Resources (Ressourcen) auf Elastic IPs (Elastic IP-Adressen) und dann auf die IP-Adresse in der Spalte Address (Adresse).
3. Klicken Sie auf der Detailseite der Adresse auf Deregister (Abmelden).



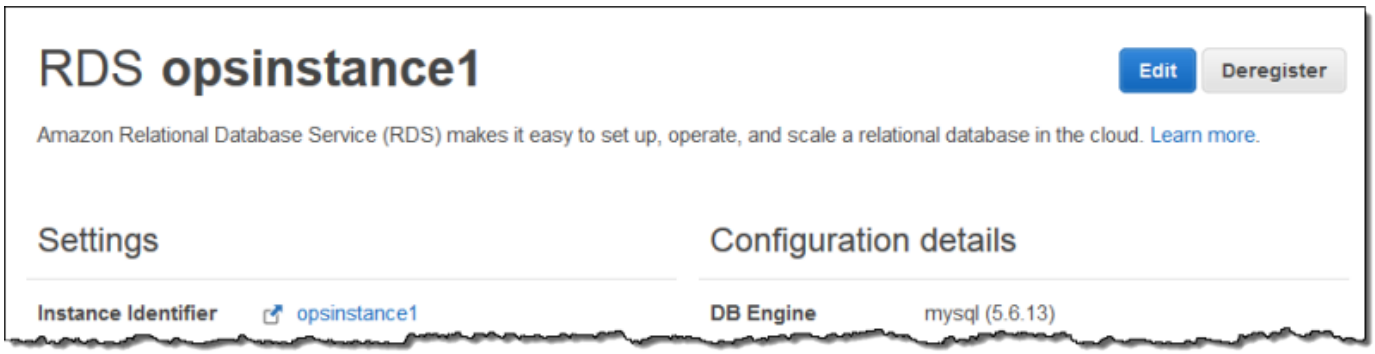
> **Note**
>
> Wenn Sie einfach eine Elastic IP-Adresse mit einem anderen Stack registrieren möchten, müssen Sie diese vom aktuellen Stack abmelden und dann wieder mit dem neuen Stack registrieren. Sie können jedoch eine zugeordnete Elastic IP-Adresse auch direkt in einen anderen Stack verschieben. Weitere Informationen finden Sie unter Zuweisen und Verschieben von Ressourcen.

Abmeldung von Amazon RDS-Instances

Gehen Sie wie folgt vor, um eine Amazon RDS-Instance zu deregistrieren.

So deregistrieren Sie eine Amazon RDS-Instance

1. Wenn die Instance einer Anwendung zugeordnet ist, trennen Sie sie, wie in Trennen von Ressourcen beschrieben.
2. Klicken Sie auf der Seite Resources (Ressourcen) auf RDS und dann auf den Namen der Instance.
3. Klicken Sie auf der Detailseite der Instance auf Deregister (Abmelden).



Tags

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Tags helfen Ihnen, Ressourcen in Chef 11.10-, Chef 12- und Chef 12.2-Stacks zu gruppieren und die Kosten der Ressourcennutzung in AWS Billing and Cost Management zu verfolgen.

Sie können auf Stack- und Layer-Ebene Tags anwenden. Wenn Sie ein Tag erstellen, wenden Sie das Tag auf alle Ressourcen innerhalb der gekennzeichneten Struktur an. Wenn Sie beispielsweise ein Tag auf eine Ebene anwenden, wenden Sie das Tag auf jede Instance, jedes Amazon EBS-Volume (außer dem Root) oder den Elastic Load Balancing Load Balancer in der Ebene an. Tags können derzeit nicht auf das Stamm- oder Standard-EBS-Volume einer Instance angewendet werden.

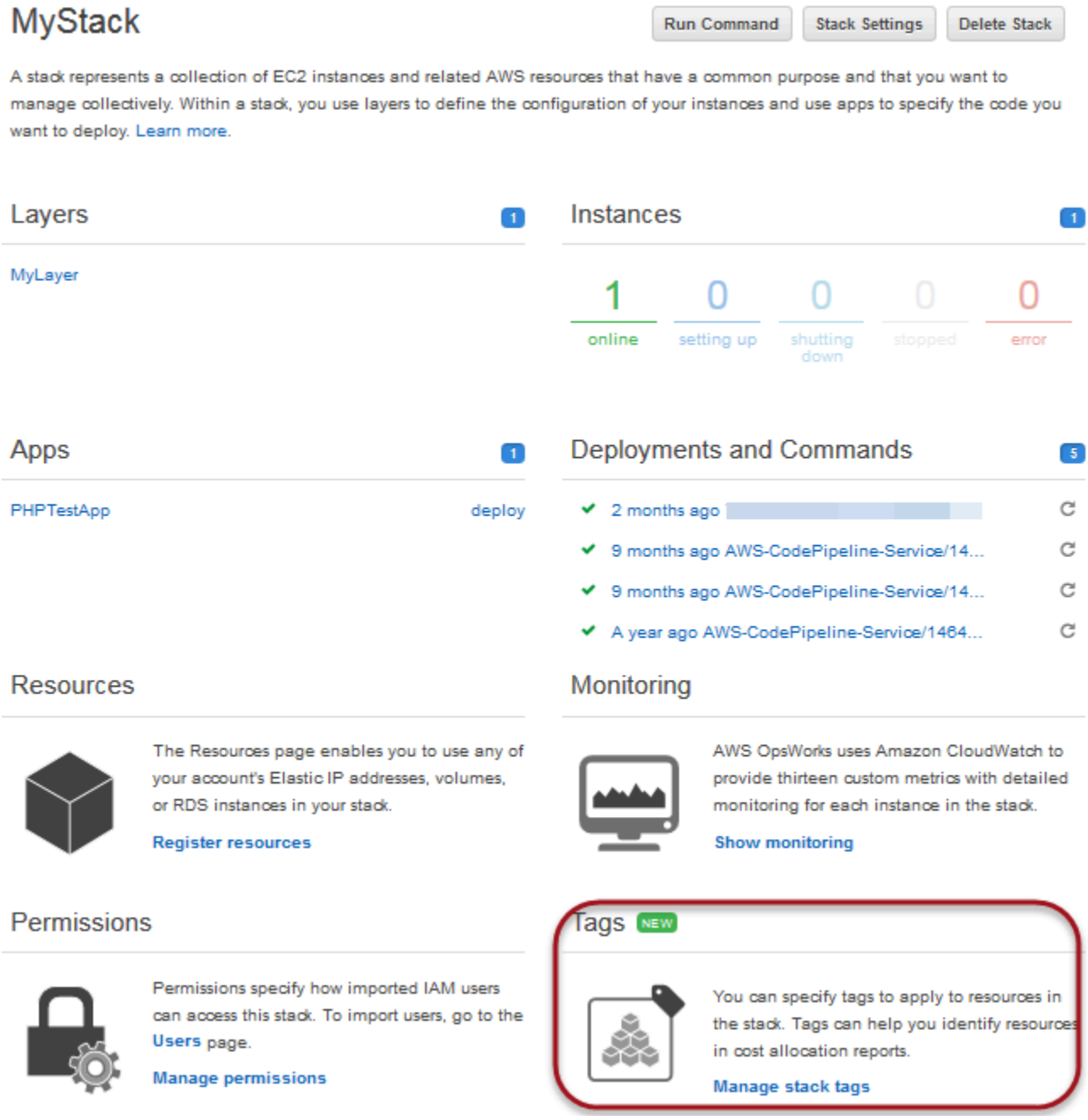
Tags sind Schlüssel-Wert-Paare, die Sie Stacks oder Ebenen in Stacks zuweisen. AWS OpsWorks Nachdem Sie Stichwörter erstellt haben, öffnen Sie die Billing and Cost Management-Konsole, um benutzerdefinierte Stichwörter zu aktivieren. Weitere Informationen dazu, wie Sie Ihre Tags aktivieren und damit die Kosten Ihrer AWS OpsWorks Stacks-Ressourcen verfolgen und verwalten können, finden Sie unter Verwenden von Kostenzuordnungs-Tags und Aktivieren von benutzerdefinierten Kostenzuordnungs-Tags im Billing and Cost Management-Benutzerhandbuch.

Tags funktionieren ähnlich wie benutzerdefinierte Attribute in AWS OpsWorks Stacks. Tags, die Sie einem Stack zuordnen, werden an jeden Layer im Stack vererbt. Auf Ebenenebene können Sie die Werte (aber nicht die Schlüsselnamen) von geerbten Tags überschreiben und neue layerspezifische Tags hinzufügen. AWS OpsWorks wendet den resultierenden Tagsatz auf alle Ressourcen in der Ebene an. Wenn Sie neue Ressourcen erstellen oder bestehende Ressourcen einem Layer zuweisen, werden die neue Ressourcen im Layer mit derselben Tag-Menge versehen.

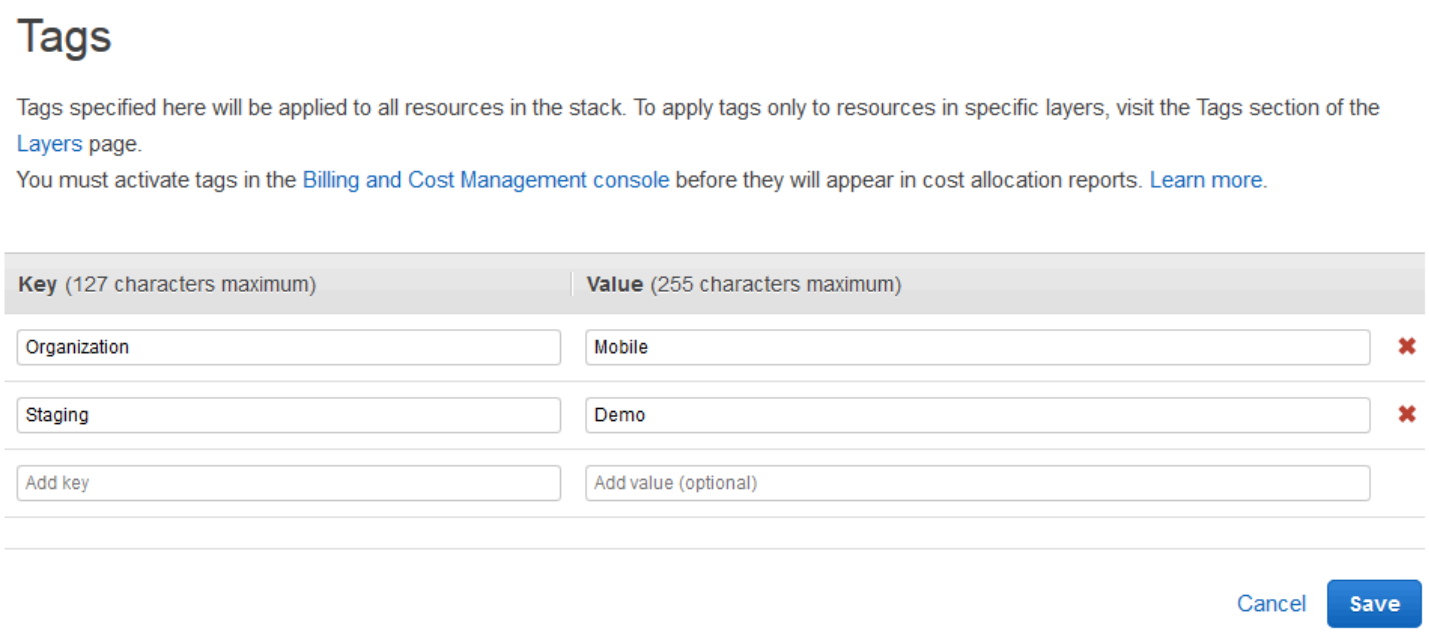
Themen

Festlegen von Tags auf der Stack-Ebene

Auf der Stack-Ebene können Sie Tags hinzufügen und verwalten, indem Sie auf der Stack-Homepage Tags auswählen.

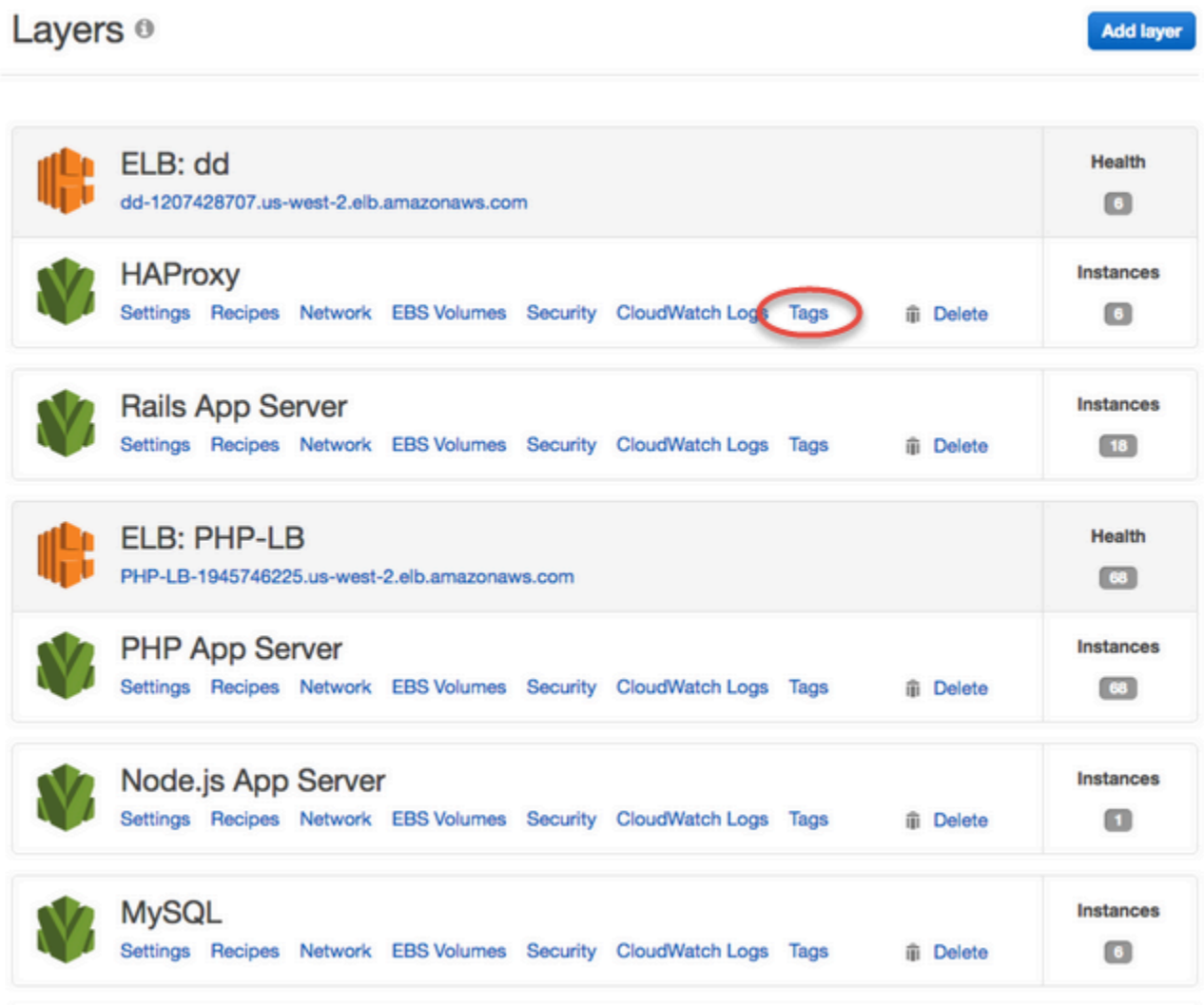


Fügen Sie auf der Seite Tags Tags als Schlüssel-Wert-Paare hinzu. Die folgende Abbildung zeigt einige Beispiel-Tags. Sie können Tags löschen, indem Sie das rote X rechts von einem Schlüssel-Wert-Paar auswählen.

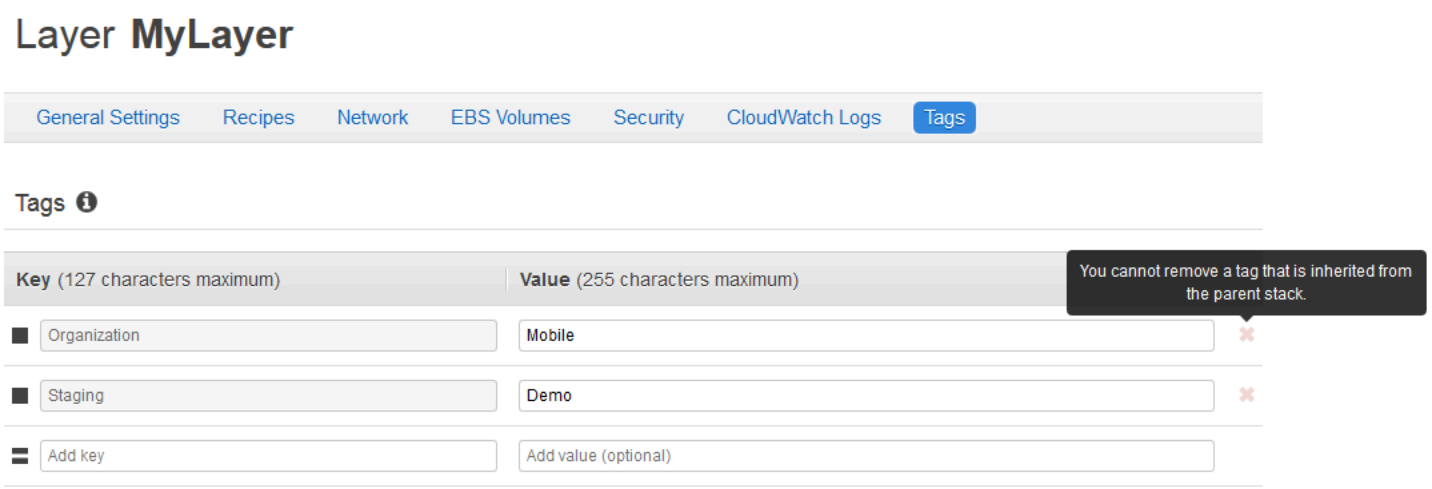


Festlegen von Tags auf der Layer-Ebene

Setzen Sie auf der Layer-Ebene Tags fest, indem Sie die Registerkarte Tags öffnen. Sie finden diese Registerkarte auf der Layers-Startseite und auf den Startseiten der einzelnen Layer.



Wenn Sie Tags auf Layer-Ebene ändern oder hinzufügen, denken Sie daran, dass Tags, die auf einer übergeordneten Ebene hinzugefügt wurden, an den Layer und dessen Ressourcen vererbt werden. Sie können die Werte vererbter Tags ändern. Sie können jedoch keine Schlüsselnamen ändern oder vererbte Tags löschen. Ändern Sie die Schlüsselnamen oder löschen Sie in den Stack-Einstellungen die von einem übergeordneten Stack geerbten Tags. Der folgende Screenshot zeigt Beispiele für Tags, die von der Stack-Ebene vererbt wurden. Vererbte Tags werden grau angezeigt.



Weitere Informationen zum Hinzufügen von Tags zu Stacks finden Sie unter Erstellen eines neuen Stacks. Weitere Informationen zum Hinzufügen von Tags zu Layers finden Sie unter Bearbeiten der Konfiguration einer Ebene OpsWorks .

Verwaltung von Tags mit dem AWS CLI

Sie können auch AWS CLI Befehle verwenden, um Tags auf Stapel- und Ebenenebene hinzuzufügen und zu entfernen. Weitere Informationen zum Herunterladen und Installieren von finden Sie unter Installation der AWS Befehlszeilenschnittstelle. AWS CLI Sie müssen Ihrem Befehl den Parameter `--region` hinzufügen, wenn der Stack, den Sie mit einem Tag versehen wollen, sich nicht in Ihrer Standardregion befindet. Layer-ARNs werden derzeit nicht in der AWS Management Console angezeigt. Führen Sie den Befehl describe-layers aus, um den ARN eines Layers zu erhalten.

Um Tags hinzuzufügen, verwenden Sie AWS CLI

- Geben Sie in der AWS CLI Befehlszeile den folgenden Befehl ein, ersetzen Sie ***Stack_or_Layer_ARN*** und geben Sie Ihre Schlüssel-Wert-Paar-Tags an, und drücken Sie dann die EINGABETASTE. Doppelte Anführungszeichen werden durch Backslashes umgangen.

  ```
  aws opsworks tag-resource --resource-arn stack_or_layer_ARN --tags "{\"key\":
  \"value\",\"key\":\"value\"}"
  ```

  Im Folgenden wird ein Beispiel gezeigt.

  ```
  aws opsworks tag-resource --resource-arn arn:aws:opsworks:us-
  east-2:800000000003:stack/500b99c0-ec00-4cgg-8a0d-1000000jjd1b --tags "{\"Stage\":
  \"Production\",\"Organization\":\"Mobile\"}"
  ```

Um Tags zu entfernen, verwenden Sie AWS CLI

- Geben Sie in der AWS CLI Befehlszeile Folgendes ein, und drücken Sie dann die EINGABETASTE.

```
aws opsworks untag-resource --resource-arn stack_or_layer_ARN --tag-keys "[\"key\",
\"key\"]"
```

Zum Entfernen von Tags geben Sie lediglich den Schlüssel des Tags an, den Sie entfernen möchten. Im Folgenden wird ein Beispiel gezeigt.

```
aws opsworks untag-resource --resource-arn arn:aws:opsworks:us-
east-2:800000000003:stack/500b99c0-ec00-4cgg-8a0d-1000000jjd1b --tag-keys "[\"Stage
\",\"Organization\"]"
```

> **Note**
>
> Sie können vererbte Tags (Tags, die auf einer übergeordneten Stack-Ebene hinzugefügt wurden) nicht aus einem Layer entfernen. Entfernen Sie stattdessen vererbte Tags aus dem Stack.

Tag-Einschränkungen

Beachten Sie beim Erstellen von Tags die folgenden Einschränkungen.

- AWS OpsWorks Stacks begrenzt die Anzahl der benutzerdefinierten Tags auf Stapel- und Ebenenebene auf 40, einschließlich benutzerdefinierter Tags, die von einer übergeordneten Ebene übernommen wurden. Somit bleiben 10 Plätze für Standard-Tags, denen vorangestellt wird, und für Tags, die von anderen Prozessen gesetzt wurden`opsworks:`, verfügbar. AWS Für eine Ressource sind maximal 50 Tags zulässig, darunter sowohl benutzerdefinierte Tags als auch Standardtags, die von erstellt wurden. AWS
- Tags dürfen nicht mit **`aws:`**, **`opsworks:`** oder **`rds:`** beginnen. Verwenden Sie **`name`** oder nicht **`Name`** als Tag-Schlüssel, da **`Name`** es von AWS OpsWorks Stacks reserviert ist.
- Ein Schlüssel darf maximal 127 Zeichen lang sein und nur Unicode-Buchstaben, Ziffern oder Trennzeichen oder die folgenden Sonderzeichen enthalten: `+ - = . _ : / .`

- Ein Wert darf maximal 255 Zeichen lang sein und nur Unicode-Buchstaben, Ziffern oder Trennzeichen oder die folgenden Sonderzeichen enthalten: `+ - = . _ : / .`

Überwachen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können Ihre Stacks folgendermaßen überwachen.

- AWS OpsWorks Stacks verwendet Amazon CloudWatch , um dreizehn benutzerdefinierte Metriken mit detaillierter Überwachung für jede Instanz im Stack bereitzustellen.
- AWS OpsWorks Stacks lässt sich integrieren AWS CloudTrail , um jeden AWS OpsWorks Stacks-API-Aufruf zu protokollieren und die Daten in einem Amazon S3 S3-Bucket zu speichern.
- Sie können Amazon CloudWatch Logs verwenden, um die System-, Anwendungs- und benutzerdefinierten Protokolle Ihres Stacks zu überwachen.

Themen

Stacks mit Amazon überwachen CloudWatch

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir

empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks verwendet Amazon CloudWatch (CloudWatch), um Stacks zu überwachen.

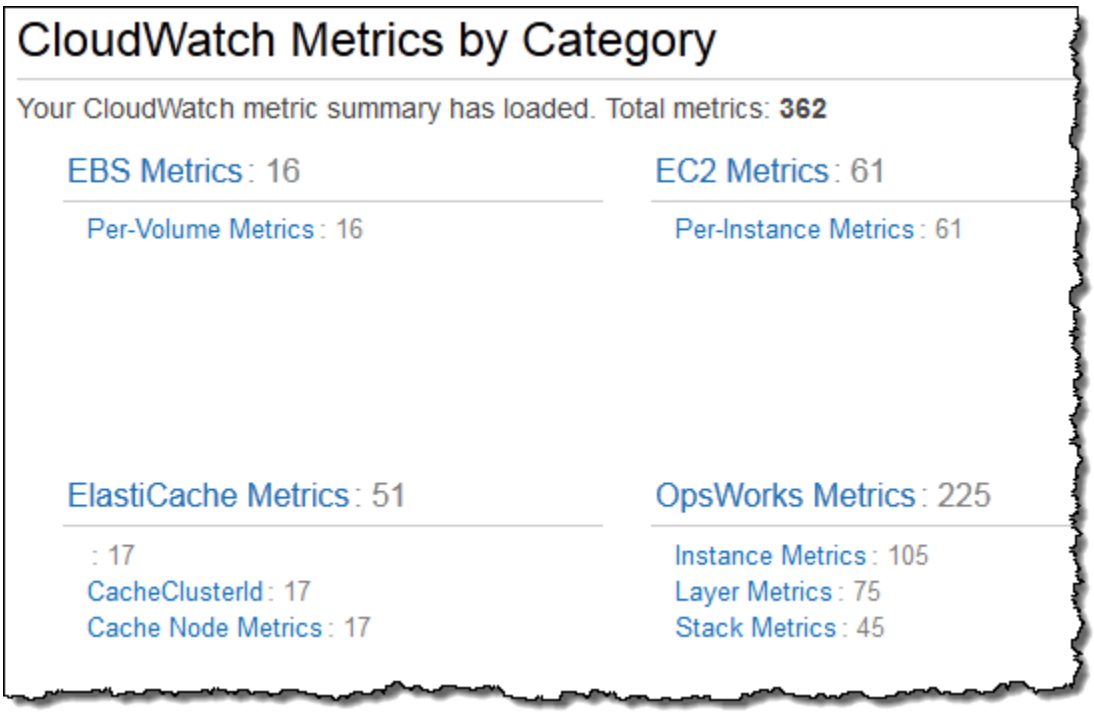
- Für Linux-Stacks unterstützt AWS OpsWorks Stacks dreizehn benutzerdefinierte Metriken, um eine detaillierte Überwachung für jede Instance im Stack zu ermöglichen, und fasst die Daten für Sie übersichtlicher auf der Monitoring-Seite zusammen.
- Für Windows-Stacks können Sie die Amazon EC2 EC2-Standardmetriken für Ihre Instances mit der CloudWatch Konsole überwachen.

  Die Seite Monitoring (Überwachung) zeigt jedoch keine Windows-Metriken an.

Auf der Monitoring-Seite werden Metriken für einen gesamten Stack, eine Ebene oder eine Instance angezeigt. AWS OpsWorks Stacks-Metriken unterscheiden sich von Amazon EC2-Metriken. Sie können auch zusätzliche Metriken über die CloudWatch Konsole aktivieren, für diese fallen jedoch in der Regel zusätzliche Gebühren an. Sie können die zugrunde liegenden Daten auch wie folgt auf der CloudWatch Konsole anzeigen:

Um OpsWorks benutzerdefinierte Metriken anzuzeigen in CloudWatch

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie auf der Navigationsleiste die Region des Stacks aus.
3. Wählen Sie im Navigationsbereich Metriken aus.
4. Wählen Sie unter OpsWorks Metriken die Optionen Instanzmetriken, Layer-Metriken oder Stack-Metriken aus.



> **Note**
>
> AWS OpsWorks Stacks sammelt Metriken, indem auf jeder Instanz (dem Instanzagenten) ein Prozess ausgeführt wird. Da Metriken mithilfe des Hypervisors unterschiedlich CloudWatch erfasst werden, können sich die Werte in der CloudWatch Konsole geringfügig von den entsprechenden Werten auf der Monitoring-Seite in der AWS OpsWorks Stacks-Konsole unterscheiden.

Sie können die CloudWatch Konsole auch verwenden, um Alarme einzustellen. Weitere Informationen zum Erstellen von Alarmen finden Sie unter CloudWatch Amazon-Alarme erstellen. Eine Liste der CloudWatch benutzerdefinierten Metriken finden Sie unter OpsWorksAWS-Metriken und -Dimensionen. Weitere Informationen finden Sie auf Amazon CloudWatch.

Themen

AWS OpsWorks Stapelt Metriken

AWS OpsWorks Stacks sendet CloudWatch alle fünf Minuten die folgenden Metriken.

CPU-Metriken

Metrik	Beschreibung
`cpu_idle`	Der Prozentsatz der Zeit, die sich die CPU im Leerlauf befindet. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder InstanceId. Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine
`cpu_nice`	Der Prozentsatz der Zeit, in der die CPU Prozesse mit einem positiven `nice` Wert verarbeitet, die eine niedrigere Scheduling-Priorität haben. Weitere Informationen darüber, was damit gemessen wird, finden Sie unter nice (Unix). Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Metriken anzeigen: StackId, LayerId, oder InstanceId. Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine
`cpu_steal`	Da AWS Hypervisor-CPU-Ressourcen auf eine zunehmende Anzahl von Instances verteilt, steigt die Virtualisierungslast, was sich darauf auswirken kann,

Metrik	Beschreibung
	wie oft der Hypervisor die angeforderte Arbeit an einer Instance ausführen kann. `cpu_steal` misst den Prozentsatz der Zeit, in der eine Instance darauf wartet, dass der Hypervisor physische CPU-Resso urcen zuweist. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder. InstanceId Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine
`cpu_system`	Der Prozentsatz der Zeit, in der die CPU Systemvor gänge verarbeitet. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder InstanceId. Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine

Metrik	Beschreibung
`cpu_user`	Der Prozentsatz der Zeit, in der die CPU Benutzeroperationen verarbeitet. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder InstanceId. Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine
`cpu_waitio`	Der Prozentsatz der Zeit, in der die CPU auf Eingabe-/Ausgabevorgänge wartet. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder. InstanceId Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine

Speichermetriken

Metrik	Beschreibung
`memory_buffers`	Die Menge des gepufferten Speichers. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder InstanceId.

Metrik	Beschreibung
	Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine
`memory_cached`	Die Menge des zwischengespeicherten Speichers. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder InstanceId. Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine
`memory_free`	Die Menge an freiem Speicher. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder InstanceId. Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine

Metrik	Beschreibung
`memory_swap`	Die Größe des Swap-Speichers. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder InstanceId. Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine
`memory_total`	Die Gesamtgröße des Speichers. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder InstanceId. Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine
`memory_used`	Die Menge des verwendeten Speichers. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder InstanceId. Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine

Metriken laden

Metrik	Beschreibung
`load_1`	Die Auslastung wurde über einen Zeitraum von einer Minute gemittelt. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder. InstanceId Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine
`load_5`	Die durchschnittliche Auslastung betrug über einen Zeitraum von fünf Minuten. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder. InstanceId Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine
`load_15`	Die Belastung wurde über einen Zeitraum von 15 Minuten gemittelt. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder InstanceId. Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`.

Metrik	Beschreibung
	Einheit: keine

Prozessmetriken

Metrik	Beschreibung
`procs`	Die Anzahl der aktiven Prozesse. Gültige Dimensionen: Die IDs der einzelnen Ressourcen, für die Sie Messwerte anzeigen: StackId, LayerId, oder InstanceId. Gültige Statistiken: `Average`,`Minimum`,`Maximum`,`Sum`, oder`Data Samples`. Einheit: keine

Dimensionen für AWS OpsWorks Stacks-Metriken

AWS OpsWorks Stacks-Metriken verwenden den AWS OpsWorks Stacks-Namespace und stellen Metriken für die folgenden Dimensionen bereit:

Dimension	Beschreibung
`StackId`	Durchschnittliche Werte für einen Stack.
`LayerId`	Durchschnittliche Werte für einen Layer.
`InstanceId`	Durchschnittliche Werte für eine Instance.

Stack-Metriken

Um eine Zusammenfassung der Metriken für einen gesamten Stack anzuzeigen, wählen Sie einen Stack im AWS OpsWorks Stacks-Dashboard aus und klicken Sie dann im Navigationsbereich auf Monitoring. Das folgende Beispiel ist für einen Stack mit einem PHP- und einem DB-Layer ausgelegt.

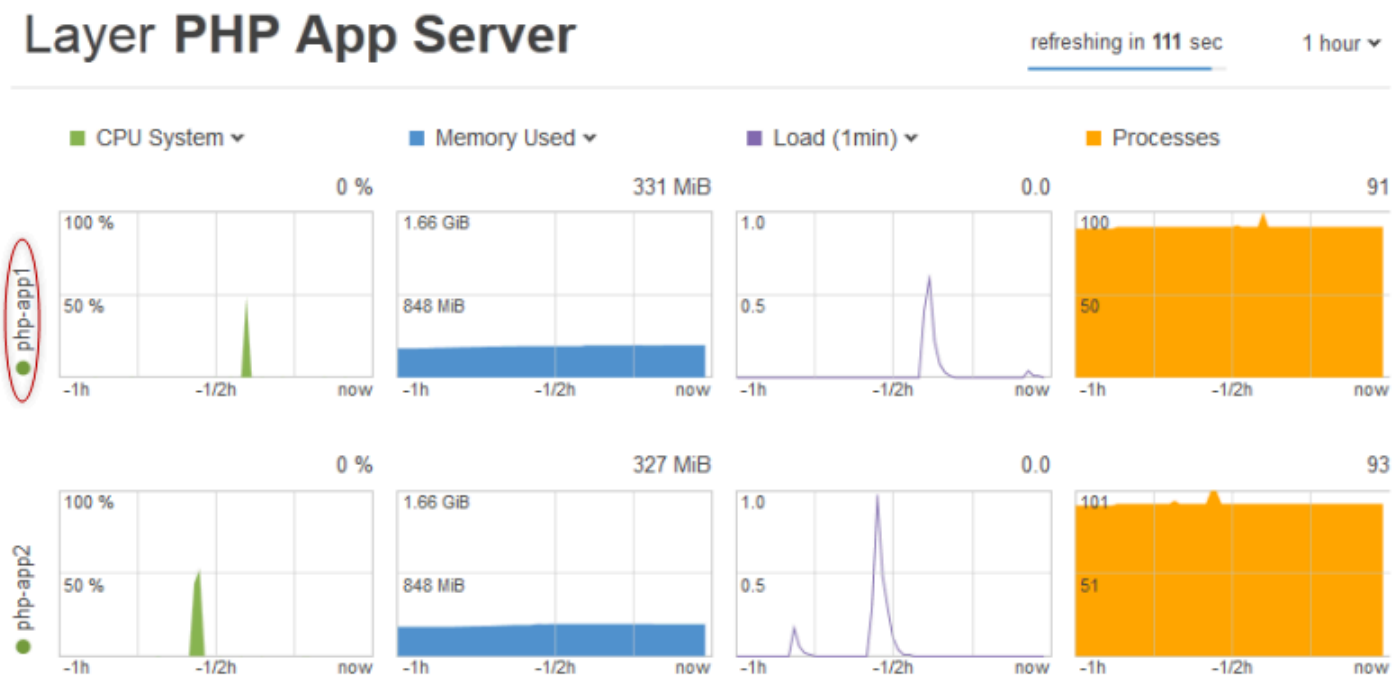
Die Stack-Ansicht zeigt für jeden Layer über einen bestimmten Zeitraum (1 Stunde, 8 Stunden, 24 Stunden, 1 Woche oder 2 Wochen) Diagramme der vier Metriktypen an. Beachten Sie Folgendes:

- AWS OpsWorks Stacks aktualisiert die Grafiken regelmäßig. Der Countdown-Timer oben rechts gibt die verbleibende Zeit bis zur nächsten Aktualisierung an.
- Wenn ein Layer mehr als eine Instance besitzt, zeigt das Diagramm Durchschnittswerte für den Layer an.
- Sie können den Zeitraum angeben, indem Sie oben rechts auf die Liste klicken und Ihren gewünschten Wert auswählen.

Für jeden Metriktyp können Sie in der Liste oben im Diagramm die Metrik auswählen, die Sie gerne anzeigen möchten.

Layer-Metriken

Wenn Sie Metriken für einen bestimmten Layer anzeigen möchten, klicken Sie auf den Layer-Namen in der Ansicht Monitoring Layers (Überwachungs-Layer). Das folgende Beispiel zeigt Metriken für den PHP-Layer, der über zwei Instances verfügt.

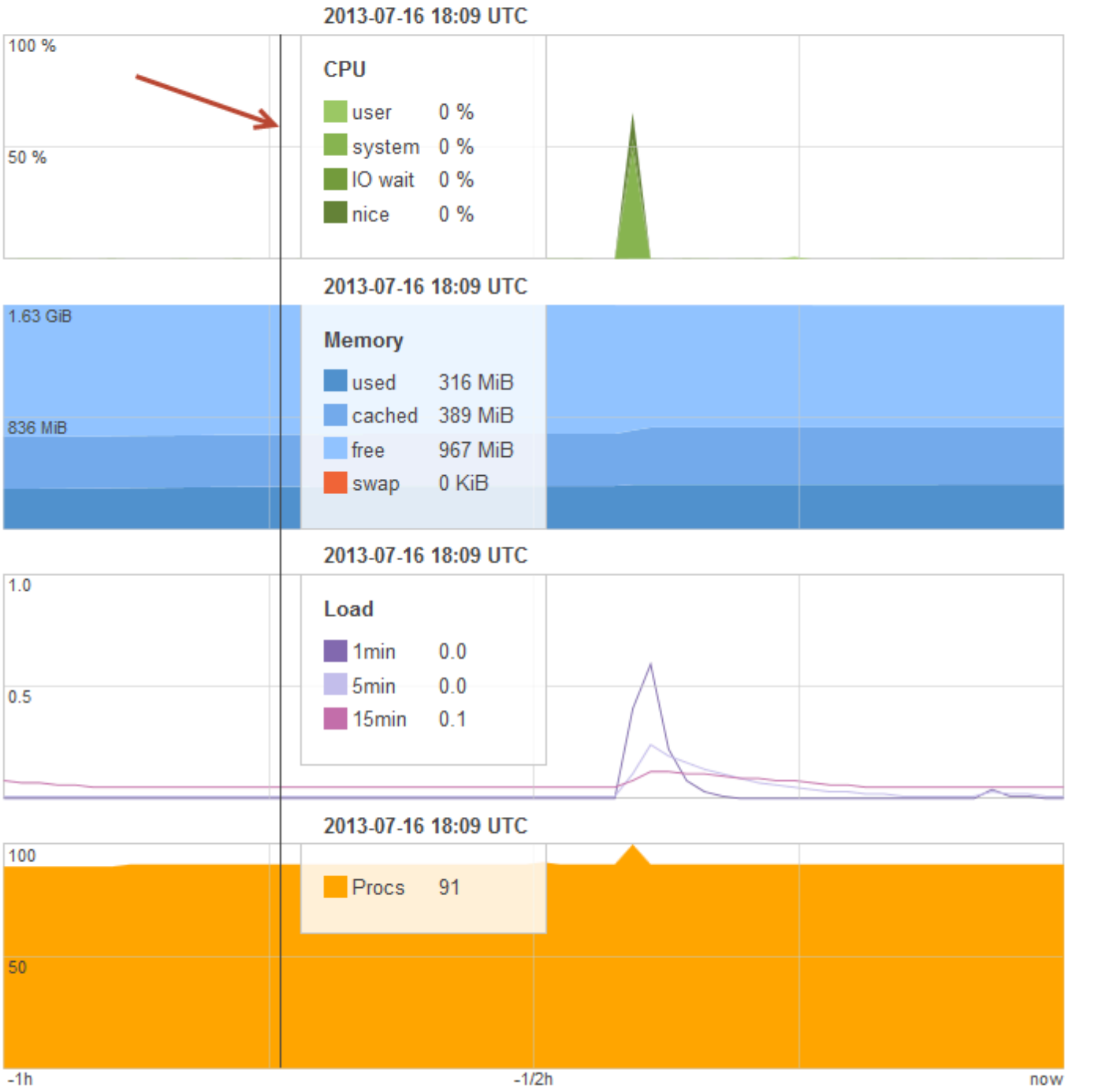


Die Metriktypen sind die gleichen wie bei den Stack-Metriken und Sie können mithilfe der List oben im Diagramm für jeden Typ die Metrikanzeige, auswählen, die Sie sehen möchten.

> **Note**
>
> Sie können auch Layer-Metriken anzeigen, indem Sie die Seite "Layer-Details" aufrufen und oben rechts auf Monitoring (Überwachung) klicken.

Instance-Metriken

Wenn Sie Metriken für eine bestimmte Instance anzeigen möchten, klicken Sie auf den Instance-Namen in der Layer-Überwachungsansicht. Das folgende Beispiel zeigt Metriken für die PHP-Layer-Instance php-app1.


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

Die Diagramme fassen alle verfügbaren Metriken für jeden Metriktyp zusammen. Um die genauen Werte für einen bestimmten Zeitpunkt abzurufen, bewegen Sie mit der Maus den Schieberegler (der rote Pfeil in der vorherigen Abbildung) auf die entsprechende Position.

> **Note**
>
> Sie können auch Instance-Metriken anzeigen, indem Sie die Detailseite der Instance aufrufen und oben rechts Monitoring (Überwachung) auswählen.

Protokollierung AWS OpsWorks von Stacks-API-Aufrufen mit AWS CloudTrail

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einer IAM-Identität oder einem AWS Dienst in Stacks ausgeführt werden. AWS OpsWorks CloudTrail erfasst alle API-Aufrufe für AWS OpsWorks Stacks als Ereignisse, einschließlich Aufrufe von der AWS OpsWorks Stacks-Konsole und von Codeaufrufen an die Stacks-APIs. AWS OpsWorks Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für AWS OpsWorks Stacks. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an AWS OpsWorks Stacks gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

AWS OpsWorks Stapelt Informationen in CloudTrail

CloudTrail ist in Ihrem AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn Aktivitäten in AWS OpsWorks Stacks auftreten, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen in der CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS Konto ansehen, suchen und herunterladen. Weitere Informationen finden Sie unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich der Ereignisse für AWS OpsWorks Stacks, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser standardmäßig für alle Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- Übersicht zum Erstellen eines Trails
- CloudTrail Unterstützte Dienste und Integrationen
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Alle AWS OpsWorks Stacks-Aktionen werden von der AWS OpsWorks Stacks-API-Referenz protokolliert CloudTrail und sind in dieser dokumentiert. Beispielsweise generieren Aufrufe der `StartInstance` Aktionen `CreateLayerDescribeInstances`, und Einträge in den CloudTrail Protokolldateien.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Gibt an, ob die Anforderung mit Root- oder IAM-Benutzer-Anmeldeinformationen ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter dem CloudTrail UserIdentity-Element.

Grundlegendes zu AWS OpsWorks Stacks-Protokolldateieinträgen

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `CreateLayer` Aktion demonstriert.


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


Amazon CloudWatch Logs mit AWS OpsWorks Stacks verwenden

Important

Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Um die Überwachung von Protokollen auf mehreren Instances zu vereinfachen, unterstützt AWS OpsWorks Stacks Amazon CloudWatch Logs. Sie aktivieren CloudWatch Logs auf Layer-Ebene in AWS OpsWorks Stacks. CloudWatch Die Log-Integration funktioniert mit den Linux-basierten Stacks Chef 11.10 und Chef 12. Wenn Sie CloudWatch Logs aktivieren, fallen zusätzliche Gebühren an. Überprüfen Sie daher die CloudWatchAmazon-Preise, bevor Sie beginnen.

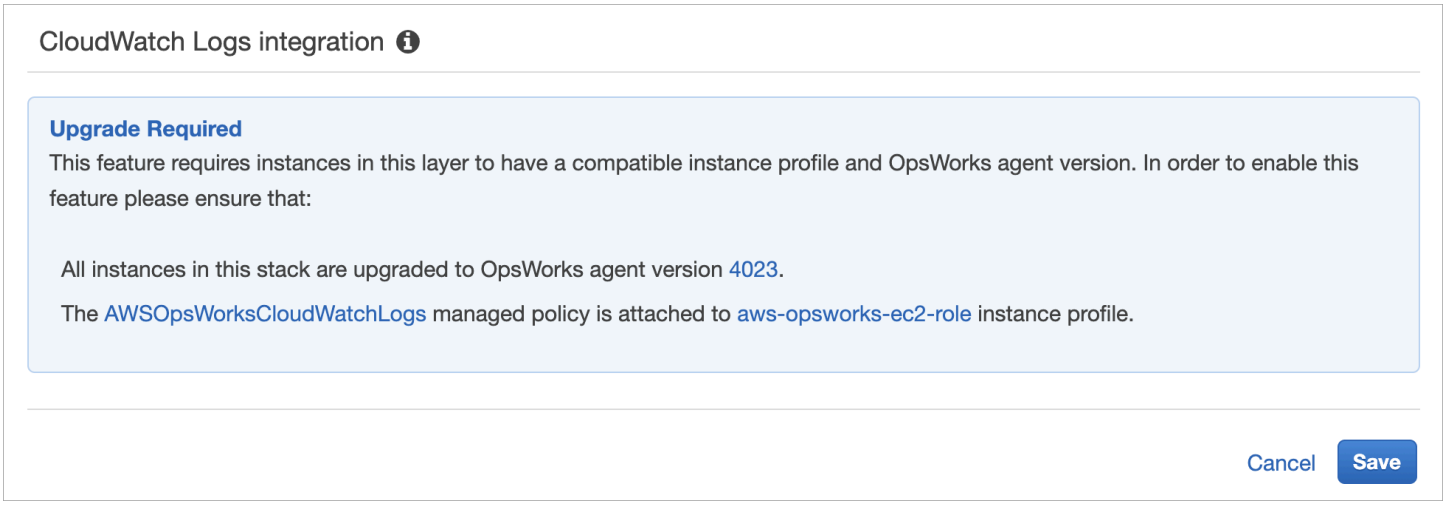
CloudWatch Logs überwacht ausgewählte Protokolle auf das Auftreten eines benutzerdefinierten Musters. Sie können die Überwachung beispielsweise auf Zeichenfolgen wie `NullReferenceException` oder die Häufigkeit solcher Ereignisse ausrichten. Nachdem Sie CloudWatch Logs in AWS OpsWorks Stacks aktiviert haben, sendet der AWS OpsWorks Stacks-Agent die Protokolle an Logs. CloudWatch Weitere Informationen zu CloudWatch Logs finden Sie unter Erste Schritte mit CloudWatch Logs.

Voraussetzungen

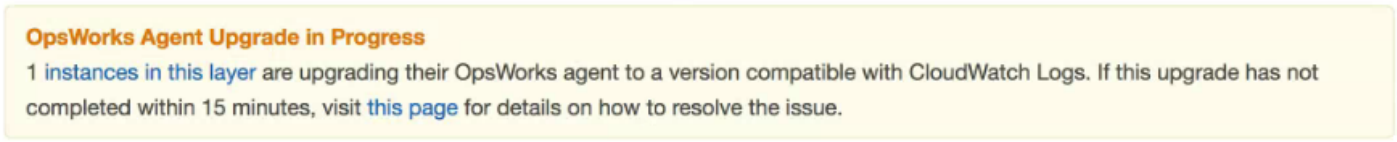
Bevor Sie CloudWatch Logs aktivieren können, müssen Ihre Instances Version 3444 oder höher des AWS OpsWorks Stacks-Agenten in Chef 11.10-Stacks und 4023 oder höher in Chef 12-Stacks ausführen. Sie müssen auch ein kompatibles Instanzprofil für alle Instanzen verwenden, die Sie mithilfe von Logs überwachen. CloudWatch

Wenn Sie ein benutzerdefiniertes Instanzprofil verwenden (eines, das AWS OpsWorks Stacks bei der Erstellung des Stacks nicht bereitgestellt hat), kann AWS OpsWorks Stacks das Instanzprofil nicht automatisch aktualisieren. Sie müssen die AWSOpsWorksCloudWatchLogsRichtlinie mithilfe von IAM manuell an Ihr Profil anhängen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Verwaltung von IAM-Richtlinien.

Wenn Sie Ihre Agentenversion oder Ihr Instanzprofil aktualisieren müssen, zeigt AWS OpsWorks Stacks eine Erinnerung an, die dem folgenden Screenshot ähnelt, wenn Sie den Tab CloudWatch Logs auf der Layer-Seite öffnen.



Es kann einige Zeit in Anspruch nehmen, den Agenten auf allen Instances eines Layers zu aktualisieren. Wenn Sie versuchen, CloudWatch Logs on a Layer zu aktivieren, bevor das Agent-Upgrade abgeschlossen ist, wird eine Meldung ähnlich der folgenden angezeigt.



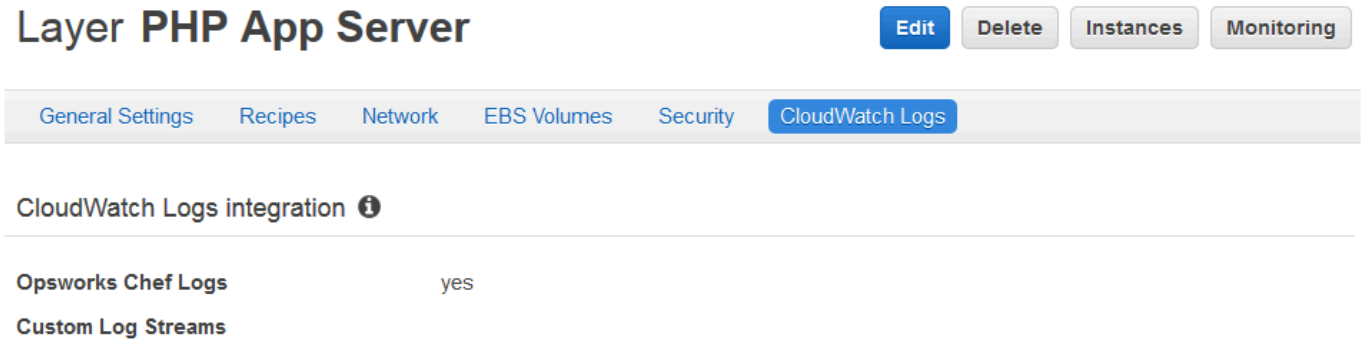
Aktivieren von CloudWatch Protokollen

1. Nachdem alle erforderlichen Upgrades von Agenten- und Instanzprofilen abgeschlossen sind, können Sie CloudWatch Logs aktivieren, indem Sie den Schieberegler auf der Registerkarte CloudWatch Logs auf On setzen.

2. Um Befehlsprotokolle zu streamen, verschieben Sie den Regler Stream command logs (Befehlsprotokolle streamen) auf On (Ein). Dadurch werden Protokolle der Chef-Aktivitäten und der vom Benutzer initiierten Befehle auf den Instanzen Ihres Layers an CloudWatch Logs gesendet.

Die in diesen Protokollen enthaltenen Daten stimmen weitgehend mit dem überein, was Sie in den Ergebnissen eines DescribeCommandsVorgangs sehen, wenn Sie das Ziel der Protokoll-URL öffnen. Es enthält Daten zu `setup`, `configure`, `deploy`, `undeploy`, `start`, `stop` sowie zu Rezeptausführungsbefehlen.

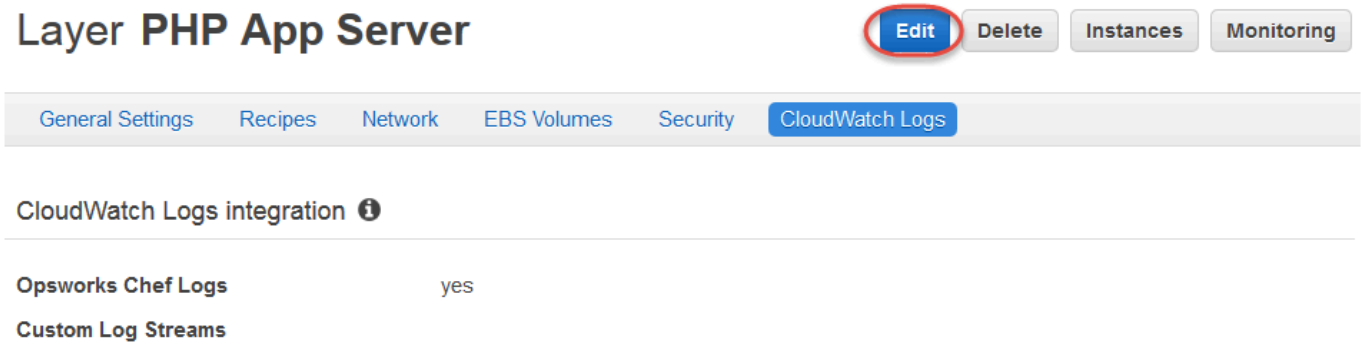
3. Um Protokolle zu Aktivitäten zu streamen, die an benutzerdefinierten Speicherorten auf den Instances des Layers gespeichert werden, z. B. `/var/log/apache/myapp/mylog*`, geben Sie den benutzerdefinierten Speicherort im Eingabefeld Stream custom logs (Befehlsprotokolle streamen) ein und klicken Sie auf Add (Hinzufügen) (+).
4. Wählen Sie Speichern. Innerhalb weniger Minuten sollten die AWS OpsWorks Stacks-Protokollstreams in der CloudWatch Logs-Konsole sichtbar sein.



CloudWatch Protokolle ausschalten

Um CloudWatch Logs zu deaktivieren, bearbeiten Sie Ihre Layer-Einstellungen.

1. Wählen Sie auf der Eigenschaftsseite des Layers Edit (Bearbeiten) aus.



2. Wählen Sie auf der Bearbeitungsseite die Registerkarte CloudWatch Protokolle aus.

3. Deaktivieren Sie im Bereich CloudWatch Logs die Option Stream-Befehlsprotokolle. Wählen Sie gegebenenfalls bei benutzerdefinierten Protokollen X aus, um die Protokolle aus den Protokoll-Streams zu löschen.
4. Wählen Sie Speichern.

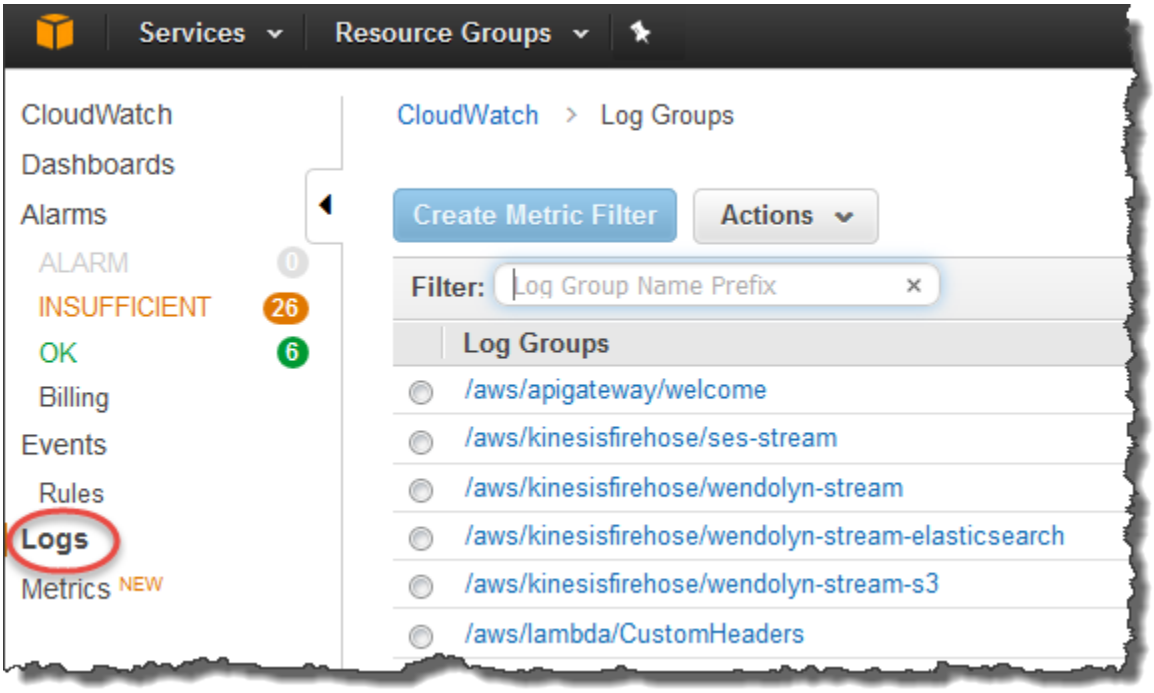
Löschen von gestreamten Protokollen aus CloudWatch Protokollen

Nachdem Sie das Streaming von CloudWatch Protokollen aus AWS OpsWorks Stacks deaktiviert haben, sind vorhandene Protokolle weiterhin in der CloudWatch Logs-Verwaltungskonsole verfügbar. Es fallen weiterhin Gebühren für gespeicherte Protokolle an, es sei denn, Sie exportieren die Protokolle nach Amazon S3 oder löschen sie. Weitere Informationen zum Exportieren von Protokollen nach S3 finden Sie unter Exportieren von Protokolldaten nach Amazon S3.

Sie können Protokollstreams und Protokollgruppen in der CloudWatch Logs-Verwaltungskonsole oder durch Ausführen der `delete-log-group` AWS CLI Befehle `delete-log-stream`und löschen. Weitere Informationen zur Änderung der Aufbewahrungsfristen für Protokolle finden Sie unter Ändern der Aufbewahrung von Protokolldaten in CloudWatch Protokollen.

Verwaltung Ihrer Logs in CloudWatch Logs

Die Logs, die Sie streamen, werden in der CloudWatch Logs-Konsole verwaltet.



AWS OpsWorks erstellt automatisch Standard-Protokollgruppen und Protokollstreams. Die Namen von Protokollgruppen für AWS OpsWorks Stacks-Daten werden nach folgendem Muster erstellt:

stack_name/*layer_name*/*chef_log_name*

Namen für benutzerdefinierte Protokolle werden nach folgendem Muster generiert:

/stack_name/layer_short_name/file_path_name. Der Pfadname wird lesbarer, wenn Sie Sonderzeichen wie "*" entfernen.

Wenn Sie Ihre Logs in CloudWatch Logs gefunden haben, können Sie die Logs in Gruppen organisieren, Logs suchen und filtern, indem Sie Metrikfilter erstellen, und benutzerdefinierte Alarme erstellen.

Konfiguration von Chef 12.2 Windows-Layern für die Verwendung von Protokollen CloudWatch

CloudWatch Die automatische Integration von Protokollen wird für Windows-basierte Instanzen nicht unterstützt. Die Registerkarte CloudWatch Protokolle ist für Ebenen in Chef 12.2-Stacks nicht verfügbar. Gehen Sie wie folgt vor, um das Streaming in CloudWatch Logs für Windows-basierte Instanzen manuell zu aktivieren.

- Aktualisieren Sie das Instanzprofil für Windows-basierte Instanzen, sodass der CloudWatch Logs-Agent über die entsprechenden Berechtigungen verfügt. Aus der AWSOpsWorksCloudWatchLogsRichtlinienerklärung geht hervor, welche Berechtigungen erforderlich sind.

  Normalerweise müssen Sie diese Aufgabe nur einmal ausführen. Sie können das aktualisierte Instance-Profil dann für alle Windows-Instances eines Layers verwenden.
- Bearbeiten Sie die folgende JSON-Konfigurationsdatei auf jeder Instance. Diese Datei enthält Einstellungen für Protokoll-Streams, beispielsweise welche Protokolle überwacht werden sollen.

  ```
  %PROGRAMFILES%\Amazon\Ec2ConfigService\Settings
  \AWS.EC2.Windows.CloudWatch.json
  ```

Sie können die beiden vorherigen Aufgaben auch automatisieren. Erstellen Sie dafür benutzerdefinierte Rezepte für die erforderlichen Aufgaben und weisen Sie sie den Setup (Einrichtung)-Ereignissen des Chef 12.2-Layers zu. Jedes Mal, wenn Sie eine neue Instanz auf

diesen Layern starten, führt AWS OpsWorks Stacks Ihre Rezepte automatisch aus, nachdem die Instanz gestartet ist, wodurch Logs aktiviert CloudWatch wird.

Um CloudWatch Logs auf Windows-basierten Instances zu deaktivieren, kehren Sie den Vorgang um. Deaktivieren Sie das Kontrollkästchen CloudWatch Logs-Integration aktivieren im Dialogfeld EC2-Diensteigenschaften, löschen Sie die Log-Stream-Einstellungen aus der `AWS.EC2.Windows.CloudWatch.json` Datei und beenden Sie die Ausführung aller Chef-Rezepte, die neuen Instanzen in Chef 12.2-Ebenen automatisch CloudWatch Logs-Berechtigungen zuweisen.

Stacks mithilfe von Amazon CloudWatch Events überwachen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können Regeln in Amazon CloudWatch Events konfigurieren, um Sie über Änderungen an den AWS OpsWorks Stacks-Ressourcen zu informieren und CloudWatch Events anzuweisen, auf der Grundlage von Veranstaltungsinhalten Maßnahmen zu ergreifen. Weitere Informationen zu den ersten Schritten mit CloudWatch Events und zum Einrichten von Regeln finden Sie unter Erste Schritte mit CloudWatch CloudWatch Events im Events-Benutzerhandbuch.

Die folgenden AWS OpsWorks Stacks-Ereignistypen werden in CloudWatch Events unterstützt.

Instance-Zustandsänderungen

Weist auf eine Änderung des Status einer AWS OpsWorks Stacks-Instanz hin.

Befehls-Zustandsänderung

Zeigt an, dass der Status eines AWS OpsWorks Stacks-Befehls geändert wurde.

Bereitstellungs-Zustandsänderung

Zeigt an, dass sich der Status einer AWS OpsWorks Stacks-Bereitstellung geändert hat.

Benachrichtigungen

Zeigt an, dass ein AWS OpsWorks Stacks-Dienstfehler ausgelöst wurde.

Weitere Informationen zu den AWS OpsWorks Stacks-Ereignistypen, die von CloudWatch Events unterstützt werden, finden Sie unter AWS OpsWorksCloudWatch Stacks-Ereignisse im Events-Benutzerhandbuch.

Sicherheit und Berechtigungen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Jeder Ihrer Benutzer muss über die entsprechenden AWS Anmeldeinformationen verfügen, um auf die Ressourcen Ihres AWS Kontos zugreifen zu können. Die empfohlene Methode zur Bereitstellung von Anmeldeinformationen für Benutzer ist AWS Identity and Access Management(IAM). AWS OpsWorks Stacks ist in IAM integriert, sodass Sie Folgendes kontrollieren können:

- Wie einzelne Benutzer mit Stacks interagieren AWS OpsWorks können.

  Sie können beispielsweise einigen Benutzern erlauben, Anwendungen in einem beliebigen Stack bereitzustellen, nicht aber, den Stack selbst zu ändern. Gleichzeitig können Sie anderen Benutzern vollen Zugriff erlauben, aber nur auf bestimmte Stacks, usw.
- Wie AWS OpsWorks Stacks in Ihrem Namen handeln kann, um auf Stack-Ressourcen wie Amazon EC2 EC2-Instances und Amazon S3 S3-Buckets zuzugreifen.

  AWS OpsWorks Stacks bietet eine Servicerolle, die Berechtigungen für diese Aufgaben gewährt.
- Wie Apps, die auf Amazon EC2 EC2-Instances ausgeführt werden, die von AWS OpsWorks Stacks gesteuert werden, auf andere AWS Ressourcen zugreifen können, z. B. auf Daten, die in Amazon S3 S3-Buckets gespeichert sind.

  Sie können den Instances eines Layers ein Instance-Profil zuweisen, das Apps, die auf diesen Instances ausgeführt werden, Berechtigungen für den Zugriff auf andere Ressourcen gewährt. AWS
- Die Art und Weise, wie benutzerbasierte SSH-Schlüssel verwaltet und SSH oder RDP zum Herstellen einer Verbindung mit Instances verwendet werden

Für jeden Stack können Administratoren jedem -Benutzer einen persönlichen SSH-Schlüssel zuweisen oder Benutzer autorisieren, ihren eigenen Schlüssel anzugeben. Sie können auch für jeden Benutzer SSH- oder RDP-Zugriff und sudo- oder Administrator-Berechtigungen auf den Instances des Stacks autorisieren.

Weitere Sicherheitsaspekte umfassen folgende Themen:

- Verwaltung der Aktualisierung des Betriebssystems Ihrer Instances mit den neuesten Sicherheits-Patches

  Weitere Informationen finden Sie unter Verwalten von Sicherheitsupdates.
- So konfigurieren Sie Amazon EC2-Sicherheitsgruppen, um den Netzwerkverkehr zu und von Ihren Instances zu kontrollieren.

  So geben Sie benutzerdefinierte Sicherheitsgruppen anstelle der Standardsicherheitsgruppen von AWS OpsWorks Stacks an. Weitere Informationen finden Sie unter Verwenden von Sicherheitsgruppen.

Themen

Verwaltung von AWS OpsWorks Stacks-Benutzerberechtigungen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Es hat sich bewährt, AWS OpsWorks Stacks-Benutzer auf einen bestimmten Satz von Aktionen oder Stack-Ressourcen zu beschränken. Sie können die Benutzerberechtigungen von AWS OpsWorks Stacks auf zwei Arten kontrollieren: indem Sie die Seite AWS OpsWorks Stacks-Berechtigungen verwenden und indem Sie eine entsprechende IAM-Richtlinie anwenden.

Auf der Seite „ OpsWorks Berechtigungen“ — oder den entsprechenden CLI- oder API-Aktionen — können Sie Benutzerberechtigungen in einer Mehrbenutzerumgebung pro Stack steuern, indem Sie jedem Benutzer eine von mehreren Berechtigungsstufen zuweisen. Jede Stufe gewährt dabei standardisierte Berechtigungen für eine Reihe von Aktionen für eine bestimmte Stack-Ressource. Auf der Seite Permissions (Berechtigungen) können Sie Folgendes festlegen:

- Wer auf einen Stack zugreifen kann
- Welche Aktionen ein Benutzer auf einem Stack ausführen kann

  Sie können bestimmten Benutzern beispielsweise nur Lesezugriff auf den Stack geben, während andere Anwendungen bereitstellen, Instances hinzufügen usw. können.
- Wer welchen Stack verwalten kann

  Sie können die Verwaltung einzelner Stacks an einen oder mehrere Benutzer übertragen
- Wer hat SSH-Zugriff und Sudo-Rechte (Linux) oder RDP-Zugriff und Administratorrechte (Windows) auf Benutzerebene für die Amazon EC2 EC2-Instances jedes Stacks.

  Sie können diese Berechtigungen jederzeit pro Benutzer gewähren oder entziehen

Important

Wenn Sie einem Benutzer keinen SSH/RDP-Zugriff gewähren, kann dieser sich möglicherweise dennoch bei Instances anmelden. Wenn Sie ein Amazon EC2 EC2-Schlüsselpaar für eine Instance angeben, kann sich jeder Benutzer mit dem entsprechenden privaten Schlüssel anmelden oder den Schlüssel verwenden, um das Windows-Administratorkennwort abzurufen. Weitere Informationen finden Sie unter Verwalten des SSH-Zugriffs.

Sie können die IAM-Konsole, CLI oder API verwenden, um Ihren Benutzern Richtlinien hinzuzufügen, die explizite Berechtigungen für die verschiedenen AWS OpsWorks Stacks-Ressourcen und -Aktionen gewähren.

- Die Verwendung einer IAM-Richtlinie zur Angabe von Berechtigungen ist flexibler als die Verwendung der Berechtigungsstufen.
- Sie können IAM-Identitäten (Benutzer, Benutzergruppen und Rollen) einrichten, die IAM-Identitäten wie Benutzern und Benutzergruppen Berechtigungen gewähren, oder Rollen definieren, die Verbundbenutzern zugeordnet werden können.
- Eine IAM-Richtlinie ist die einzige Möglichkeit, Berechtigungen für bestimmte wichtige Stacks-Aktionen zu gewähren. AWS OpsWorks

  Beispielsweise müssen Sie IAM verwenden, um Berechtigungen für `opsworks:CreateStack` und zu erteilen`opsworks:CloneStack`, die jeweils zum Erstellen und Klonen von Stacks verwendet werden.

Es ist zwar nicht explizit möglich, Verbundbenutzer in die Konsole zu importieren, aber ein Verbundbenutzer kann implizit ein Benutzerprofil erstellen, indem er oben rechts in der AWS OpsWorks Stacks-Konsole Meine Einstellungen und dann ebenfalls oben rechts Benutzer auswählt. Auf der Seite Benutzer können Verbundbenutzer, deren Konten mithilfe der API oder CLI oder implizit über die Konsole erstellt wurden, ihre Konten ähnlich wie Benutzer ohne Verbundbenutzer verwalten.

Beide Methoden schließen sich nicht gegenseitig aus und lassen sich in einigen Fällen sogar sinnvoll kombinieren. AWS OpsWorks Stacks wertet dann beide Berechtigungen aus. Angenommen, Sie möchten Benutzern die Berechtigung zum Hinzufügen oder Löschen von Instances, nicht jedoch von Layers gewähren. Keine der Stacks-Berechtigungsstufen gewährt diesen bestimmten Satz von Berechtigungen AWS OpsWorks . Sie können jedoch die Seite „Berechtigungen“ verwenden, um Benutzern die Berechtigungsstufe Verwalten zu gewähren, mit der sie die meisten Stack-Operationen ausführen können, und dann eine IAM-Richtlinie anwenden, die Berechtigungen zum Hinzufügen oder Entfernen von Layern verweigert. Weitere Informationen finden Sie unter Steuern des Zugriffs auf AWS Ressourcen mithilfe von Richtlinien.

Nachfolgend finden Sie ein Modell zur Verwaltung von Benutzerberechtigungen. Es wird in jedem Fall davon ausgegangen, dass der Leser (Sie), ein administrativer Benutzer ist.

1. Verwenden Sie die IAM-Konsole, um AWSOpsWorks_FullAccess Richtlinien auf einen oder mehrere Administratorbenutzer anzuwenden.

2. Erstellen Sie für jeden Benutzer ohne Administratorrechte einen Benutzer mit einer Richtlinie, die keine AWS OpsWorks Stacks-Berechtigungen gewährt.

   Wenn ein Benutzer nur Zugriff auf AWS OpsWorks Stacks benötigt, müssen Sie möglicherweise überhaupt keine Richtlinie anwenden. Stattdessen können Sie ihre Berechtigungen auf der Seite AWS OpsWorks Stacks-Berechtigungen verwalten.
3. Verwenden Sie die Seite AWS OpsWorks Stacks-Benutzer, um Benutzer ohne Administratorrechte in Stacks zu importieren. AWS OpsWorks
4. Weisen Sie auf der Seite Permissions (Berechtigungen) des jeweiligen Stacks den einzelnen Benutzern Berechtigungsebenen zu.
5. Passen Sie bei Bedarf die Berechtigungsstufen der Benutzer an, indem Sie eine entsprechend konfigurierte IAM-Richtlinie anwenden.

Weitere Empfehlungen zur Verwaltung von Benutzern finden Sie unterBewährte Methoden: Verwalten von Berechtigungen.

Weitere Informationen zu bewährten Methoden für IAM finden Sie unter Bewährte Sicherheitsmethoden in IAM im IAM-Benutzerhandbuch.

Themen

Stacks-Benutzer verwalten AWS OpsWorks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Bevor du Benutzer in AWS OpsWorks Stacks importieren und ihnen Berechtigungen erteilen kannst, musst du zunächst für jede Person einen Benutzer erstellt haben. Um IAM-Benutzer zu erstellen, melden Sie sich zunächst AWS als Benutzer an, dem die in der FullAccess IAM-Richtlinie definierten Berechtigungen erteilt wurden. Anschließend verwenden Sie die IAM-Konsole, um IAM-Benutzer für alle zu erstellen, die auf Stacks zugreifen müssen. AWS OpsWorks Anschließend können Sie diese Benutzer wie folgt in AWS OpsWorks Stacks importieren und Benutzerberechtigungen gewähren:

Reguläre AWS OpsWorks Stacks-Benutzer

Reguläre Benutzer benötigen keine angehängte Richtlinie. Wenn sie über eine verfügen, beinhaltet diese in der Regel keine AWS OpsWorks Stacks-Berechtigungen. Verwenden Sie stattdessen die Seite AWS OpsWorks Stacks-Berechtigungen, um regulären Benutzern auf Basis einer der folgenden Berechtigungsstufen zuzuweisen. stack-by-stack

- Mit der Berechtigung Show (Anzeigen) können Benutzer den Stack betrachten, aber keine Operationen auf dem Stack ausführen.
- Die Berechtigung Deploy (Bereitstellen) umfasst die Berechtigung Show (Anzeigen) und ermöglicht es Benutzern außerdem, Apps bereitzustellen und zu aktualisieren.
- Die Berechtigung Manage (Verwalten) umfasst die Berechtigung Deploy (Bereitstellen) und gewährt dem Benutzer darüber hinaus die Berechtigung zur Verwaltung des Stacks mit Aufgaben wie dem Hinzufügen von Layers oder Instances. Richten Sie die Benutzerberechtigungen auf der Seite Permissions (Berechtigungen) ein und aktivieren Sie für jeden Benutzer einzeln nach Bedarf SSH/RDP-Zugriff sowie sudo-/Administratorberechtigungen.
- Mit der Berechtigung Deny (Verweigern) verweigern Sie den Zugriff auf den Stack.

Wenn diese Berechtigungsstufen nicht ganz Ihren Wünschen für einen bestimmten Benutzer entsprechen, können Sie die Berechtigungen des Benutzers anpassen, indem Sie eine IAM-Richtlinie anwenden. Sie könnten beispielsweise die Seite „ AWS OpsWorks Stacks-Berechtigungen“ verwenden, um einem Benutzer die Berechtigungsstufe „Verwalten“ zuzuweisen, wodurch er berechtigt ist, alle Stack-Management-Operationen durchzuführen, aber nicht, Stacks zu erstellen oder zu klonen. Sie könnten dann eine Richtlinie anwenden, die diese Berechtigungen einschränkt, indem sie ihnen die Erlaubnis verweigert, Ebenen hinzuzufügen oder zu löschen, oder diese Berechtigungen erweitert, indem sie ihnen erlaubt, Stacks zu erstellen oder zu klonen. Weitere Informationen finden Sie unter Verwaltung von AWS OpsWorks Stacks-Berechtigungen durch Anhängen einer IAM-Richtlinie.

AWS OpsWorks Stacks-Benutzer mit Administratorrechten

Administratorbenutzer sind der Kontoinhaber oder ein IAM-Benutzer mit den in der Richtlinie definierten Berechtigungen. AWSOpsWorks_FullAccess Neben den Berechtigungen der Berechtigungsebene Manage (Verwalten) verleiht diese Richtlinie Berechtigungen für Aktionen, die Sie auf der Seite Permissions (Berechtigungen) nicht gewähren können, darunter folgende:

- Benutzer in AWS OpsWorks Stacks importieren
- Erstellen und Klonen von Stacks

Eine vollständige Beschreibung der Richtlinie finden Sie unter Beispielrichtlinien. Eine ausführliche Liste der Berechtigungen, die Benutzern nur durch Anwendung einer IAM-Richtlinie gewährt werden können, finden Sie unter. AWS OpsWorks Stapelt die Berechtigungsstufen

Themen

- Benutzer und Regionen
- Einen AWS OpsWorks Stacks-Administratorbenutzer erstellen
- IAM-Benutzer für Stacks erstellen AWS OpsWorks
- Benutzer in Stacks importieren AWS OpsWorks
- Stacks-Benutzereinstellungen bearbeiten AWS OpsWorks

Benutzer und Regionen

AWS OpsWorks Stacks-Benutzer sind innerhalb des regionalen Endpunkts verfügbar, auf dem sie erstellt wurden. Sie können Benutzer in jeder der folgenden Regionen erstellen.

- Region USA Ost (Ohio)
- Region USA Ost (Nord-Virginia)
- Region USA West (Oregon)
- Region US West (N. California)
- Region Kanada (Zentral) (nur API); nicht verfügbar in AWS Management Console
- Region Asien-Pazifik (Mumbai)
- Region Asien-Pazifik (Singapur)
- Region Asien-Pazifik (Sydney)
- Region Asien-Pazifik (Tokio)

- Region Asien-Pazifik (Seoul)
- Region Europa (Frankfurt)
- Region Europa (Irland)
- Region Europa (London)
- Region Europa (Paris)
- Region Südamerika (São Paulo)

Wenn Sie Benutzer in AWS OpsWorks Stacks importieren, importieren Sie sie an einen der regionalen Endpunkte. Wenn Sie möchten, dass ein Benutzer in mehr als einer Region verfügbar ist, müssen Sie den Benutzer in diese Region importieren. Sie können AWS OpsWorks Stacks-Benutzer auch aus einer Region in eine andere importieren. Wenn Sie einen Benutzer in eine Region importieren, in der es bereits einen Benutzer mit demselben Namen gibt, ersetzt der importierte Benutzer den vorhandenen Benutzer. Weitere Informationen zum Importieren von Benutzern finden Sie unter Importieren von Benutzern.

Einen AWS OpsWorks Stacks-Administratorbenutzer erstellen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können einen AWS OpsWorks Stacks-Administratorbenutzer erstellen, indem Sie die `AWSOpsWorks_FullAccess` Richtlinie einem Benutzer hinzufügen, wodurch diesem Benutzer AWS OpsWorks Stacks-Vollzugriffsberechtigungen gewährt werden. Weitere Informationen zum Erstellen eines Administratorbenutzers finden Sie unter Administratorbenutzer erstellen.

> **Note**
>
> Die AWSOpsWorks_FullAccess Richtlinie ermöglicht es Benutzern, AWS OpsWorks Stacks-Stacks zu erstellen und zu verwalten, aber Benutzer können keine IAM-Dienstrolle für den Stack erstellen. Sie müssen eine vorhandene Rolle verwenden. Der erste Benutzer, der einen Stack erstellt, muss über zusätzliche IAM-Berechtigungen verfügen, wie unter beschrieben.

Administrative Berechtigungen Wenn dieser Benutzer den ersten Stack erstellt, erstellt AWS OpsWorks Stacks eine IAM-Servicerolle mit den erforderlichen Berechtigungen. Danach kann jeder Benutzer mit der Berechtigung `opsworks:CreateStack` diese Rolle verwenden, um weitere Stacks zu erstellen. Weitere Informationen finden Sie unter AWS OpsWorks Stacks erlauben, in Ihrem Namen zu handeln.

Wenn Sie einen Benutzer erstellen, können Sie zusätzliche vom Kunden verwaltete Richtlinien hinzufügen, um die Berechtigungen des Benutzers nach Bedarf zu optimieren. Dies kann beispielsweise hilfreich sein, wenn administrative Benutzer zwar in der Lage sein sollen, Stacks zu erstellen und zu löschen, nicht jedoch neue Benutzer zu importieren. Weitere Informationen finden Sie unter Verwaltung von AWS OpsWorks Stacks-Berechtigungen durch Anhängen einer IAM-Richtlinie.

Wenn Sie mehrere Administratorbenutzer haben, können Sie die AWSOpsWorks_FullAccess Richtlinie einer IAM-Gruppe hinzufügen und die Benutzer dieser Gruppe hinzufügen, anstatt die Berechtigungen für jeden Benutzer separat festzulegen.

Informationen zum Erstellen einer Gruppe finden Sie unter IAM-Benutzergruppen erstellen. Wenn Sie die Gruppe erstellen, fügen Sie die AWSOpsWorks_FullAccessRichtlinie hinzu. Sie können auch die AdministratorAccessRichtlinie hinzufügen, die die AWSOpsWorks_FullAccessBerechtigungen enthält.

Informationen zum Hinzufügen von Berechtigungen zu einer vorhandenen Gruppe finden Sie unter Eine Richtlinie an eine IAM-Benutzergruppe anhängen.

IAM-Benutzer für Stacks erstellen AWS OpsWorks

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Bevor Sie IAM-Benutzer in AWS OpsWorks Stacks importieren können, müssen Sie sie erstellen. Sie können dies über die IAM-Konsole, die Befehlszeile oder die API tun. Vollständige Anweisungen finden Sie unter Einen IAM-Benutzer in Ihrem AWS Konto erstellen.

Im Gegensatz zu administrativen Benutzern müssen Sie hier keine Richtlinie anfügen, um Berechtigungen zu verleihen. Sie können die Berechtigungen festlegen, nachdem Sie die Benutzer in AWS OpsWorks Stacks importiert haben, wie in Verwalten von Benutzerberechtigungen beschrieben.

Weitere Informationen zum Erstellen von IAM-Benutzern und -Gruppen finden Sie unter Erste Schritte mit IAM.

Benutzer in Stacks importieren AWS OpsWorks

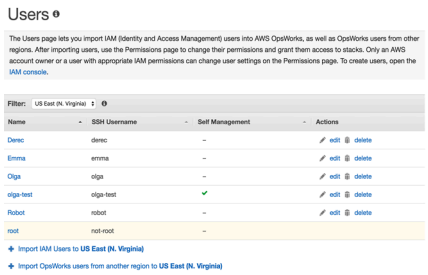
> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Administratorbenutzer können Benutzer in AWS OpsWorks Stacks importieren. Sie können auch Stacks-Benutzer von einem regionalen AWS OpsWorks Endpunkt auf einen anderen importieren. Wenn Sie Benutzer in AWS OpsWorks Stacks importieren, importieren Sie sie auf einen der regionalen AWS OpsWorks Stacks-Endpunkte. Wenn Sie möchten, dass ein Benutzer in mehr als einer Region verfügbar ist, müssen Sie den Benutzer in diese Region importieren.

Es ist zwar nicht explizit möglich, Verbundbenutzer in die Konsole zu importieren, aber ein Verbundbenutzer kann implizit ein Benutzerprofil erstellen, indem er oben rechts in der AWS OpsWorks Stacks-Konsole Meine Einstellungen und dann ebenfalls oben rechts Benutzer auswählt. Auf der Seite Benutzer können Verbundbenutzer, deren Konten mithilfe der API oder CLI oder implizit über die Konsole erstellt wurden, ihre Konten ähnlich wie Benutzer ohne Verbundbenutzer verwalten.

Um AWS OpsWorks Benutzer in Stacks zu importieren

1. Melden Sie sich als Administratorbenutzer oder als Kontoinhaber bei AWS OpsWorks Stacks an.
2. Wählen Sie oben rechts Users (Benutzer) aus, um die Seite Users (Benutzer) aufzurufen.



3. Wählen Sie IAM-Benutzer importieren in **< *Regionsname* >**, um die Benutzer anzuzeigen, die verfügbar sind, aber noch nicht importiert wurden.

4. Aktivieren Sie das Kontrollkästchen Select all (Alle auswählen) oder wählen Sie Benutzer einzeln aus. Wenn Sie fertig sind, wählen Sie „Importieren nach OpsWorks“.

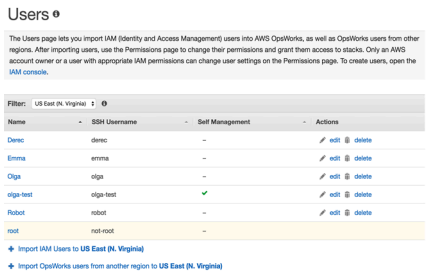
> **Note**
>
> Wenn Sie nach dem Import eines Benutzers in AWS OpsWorks Stacks die IAM-Konsole oder API verwenden, um den Benutzer aus Ihrem Konto zu löschen, verliert der Benutzer nicht automatisch den SSH-Zugriff, den Sie über Stacks gewährt haben. AWS OpsWorks Sie müssen den Benutzer auch aus AWS OpsWorks Stacks löschen, indem Sie die Seite Benutzer öffnen und in der Spalte Aktionen des Benutzers die Option Löschen auswählen.

Um AWS OpsWorks Stacks-Benutzer aus einer Region in eine andere zu importieren

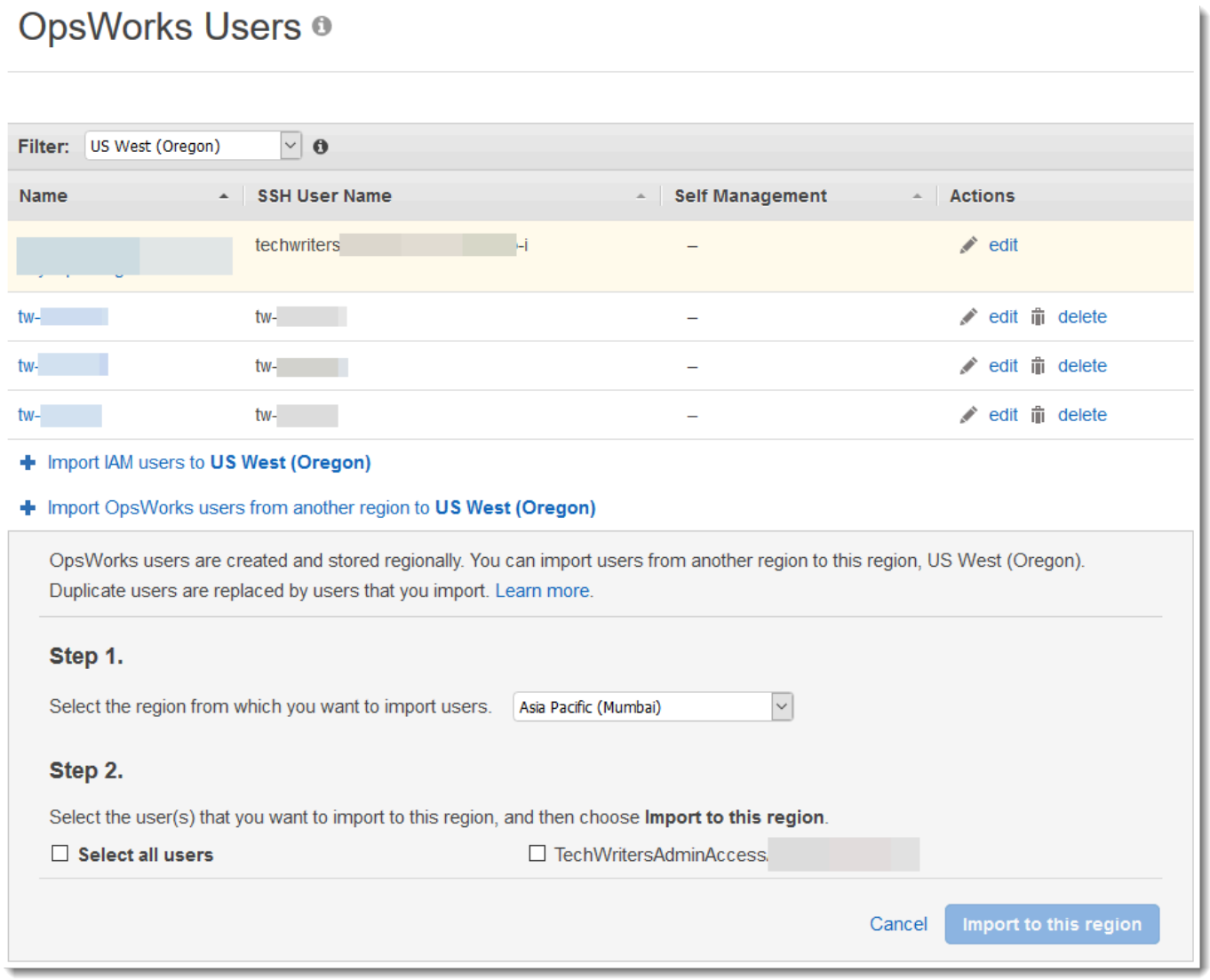
AWS OpsWorks Stacks-Benutzer sind innerhalb des regionalen Endpunkts verfügbar, auf dem sie erstellt wurden. Sie können Benutzer in den Regionen erstellen, die unter angezeigt werden. Benutzer und Regionen

Sie können AWS OpsWorks Stacks-Benutzer aus einer Region in die Region importieren, nach der Ihre Benutzerliste derzeit gefiltert ist. Wenn Sie einen Benutzer in eine Region importieren, in der es bereits einen Benutzer mit demselben Namen gibt, ersetzt der importierte Benutzer den vorhandenen Benutzer.

1. Melden Sie sich als Administratorbenutzer oder als Kontoinhaber bei AWS OpsWorks Stacks an.
2. Wählen Sie oben rechts Users (Benutzer) aus, um die Seite Users (Benutzer) aufzurufen. Wenn Sie AWS OpsWorks Stacks-Benutzer in mehr als einer Region haben, verwenden Sie das Steuerelement Filter, um nach der Region zu filtern, in die Sie Benutzer importieren möchten.



3. Wählen Sie AWS OpsWorks Stacks-Benutzer aus einer anderen Region in **< *aktuelle Region* importieren >**.



4. Wählen Sie die Region aus, aus der Sie AWS OpsWorks Stacks-Benutzer importieren möchten.
5. Wählen Sie die zu importierenden Benutzer oder alle Benutzer aus und wählen Sie Import to this region (In diese Region importieren) aus. Warten Sie, bis AWS OpsWorks Stacks die importierten Benutzer in der Benutzerliste anzeigt.

Unix-IDs und Benutzer, die außerhalb AWS OpsWorks von Stacks erstellt wurden

AWS OpsWorks weist Benutzern auf AWS OpsWorks Stacks-Instanzen Unix-ID-Werte (UID) zwischen 2000 und 4000 zu. Da der UID-Bereich zwischen 2000 und 4000 AWS OpsWorks reserviert ist, können Benutzer, die Sie außerhalb von erstellen AWS OpsWorks (z. B. mithilfe von Kochbuchrezepten oder durch Import von Benutzern AWS OpsWorks aus IAM), UIDs haben, die von Stacks für einen anderen Benutzer überschrieben werden. AWS OpsWorks Dies kann dazu führen, dass Benutzer, die Sie außerhalb von AWS OpsWorks Stacks erstellt haben, nicht in den Suchergebnissen für Datenbeutel angezeigt werden oder vom integrierten Stacks-Vorgang ausgeschlossen werden. AWS OpsWorks `sync_remote_users`

Externe Prozesse können auch Benutzer mit UIDs erstellen, die AWS OpsWorks Stacks überschreiben kann. Beispielsweise können einige Betriebssystem-Pakete einen Benutzer innerhalb von Prozessen nach der Installation erstellen. *Wenn Sie oder ein Softwareprozess einen Benutzer auf einem Linux-basierten Betriebssystem erstellen, ohne explizit eine UID anzugeben – was die Standardeinstellung ist –, lautet die von Stacks zugewiesene UID <höchste existierende UID> + 1. AWS OpsWorks AWS OpsWorks*

Es hat sich bewährt, AWS OpsWorks Stacks-Benutzer zu erstellen und deren Zugriff in der Stacks-Konsole oder mithilfe eines SDK zu verwalten. AWS OpsWorks AWS CLI AWS Wenn Sie Benutzer auf AWS OpsWorks Stacks-Instanzen außerhalb von erstellen AWS OpsWorks, verwenden Sie *UnixID-Werte* über 4000.

Stacks-Benutzereinstellungen bearbeiten AWS OpsWorks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Nachdem Sie Benutzer importiert haben, können Sie die Benutzereinstellungen wie folgt bearbeiten:

So bearbeiten Sie Benutzereinstellungen

1. Wählen Sie auf der Seite Users (Benutzer) die Option Edit (Bearbeiten) in der Spalte Actions (Aktionen) des Benutzers aus.
2. Sie können die folgenden Einstellungen festlegen.

   Selbstverwaltung

   Wählen Sie Ja, damit der Benutzer die MySettings Seite verwenden kann, um seinen persönlichen SSH-Schlüssel anzugeben.

   > **Note**
   >
   > Sie können die Selbstverwaltung auch aktivieren, indem Sie der IAM-Identität eine IAM-Richtlinie hinzufügen, die Berechtigungen für die Aktionen und gewährt. DescribeMyUserProfileUpdateMyUserProfile

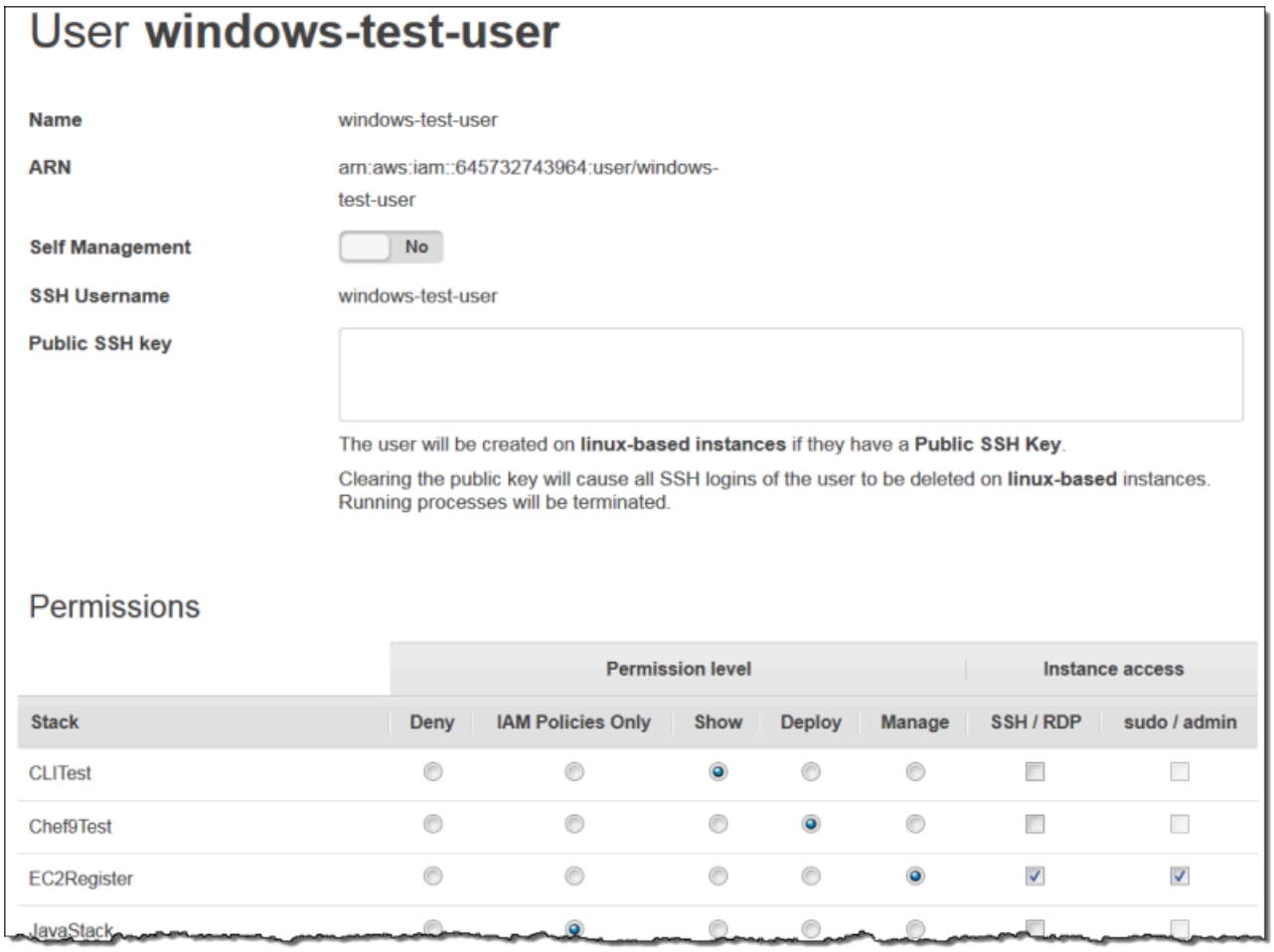
   Öffentlicher SSH-Schlüssel

   (Optional) Geben Sie einen öffentlichen SSH-Schlüssel für den Benutzer ein. Dieser Schlüssel wird auf der Seite My Settings (Eigene Einstellungen) des Benutzers angezeigt. Wenn Sie die Selbstverwaltung aktivieren, kann der Benutzer My Settings (Eigene Einstellungen) bearbeiten und einen eigenen Schlüssel eingeben. Weitere Informationen finden Sie unter Registrierung des öffentlichen SSH-Schlüssels eines Benutzers.

   AWS OpsWorks Stacks installiert diesen Schlüssel auf allen Linux-Instances. Benutzer können sich mit dem zugehörigen privaten Schlüssel anmelden. Weitere Informationen finden Sie unter Anmelden mit SSH. Dieser Schlüssel kann nicht auf Windows-Stacks eingesetzt werden.

   Berechtigungen

   (Optional) Legen Sie zentral Berechtigungsebenen für den Benutzer für die einzelnen Stacks fest, statt sie auf der Seite Permissions (Berechtigungen) jedes einzelnen Stacks festzulegen. Weitere Informationen zu Berechtigungsebenen finden Sie unter Verleihen von Berechtigungen pro Stack.



Erteilen von AWS OpsWorks Stack-Benutzerberechtigungen pro Stack

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Die einfachste Möglichkeit, die Benutzerberechtigungen von AWS OpsWorks Stacks zu verwalten, besteht darin, die Berechtigungsseite eines Stacks zu verwenden. Jeder Stack hat eine eigene Seite, auf der Berechtigungen für diesen Stack verliehen werden.

Um Berechtigungseinstellungen bearbeiten zu können, müssen Sie als administrativer Benutzer oder als Benutzer mit der Berechtigung Manage (Verwalten) angemeldet sein. Die Liste zeigt nur die Benutzer, die in AWS OpsWorks Stacks importiert wurden. Weitere Informationen zum Erstellen und Importieren von Benutzern finden Sie unter Verwalten von Benutzern.

Die Standardberechtigungsstufe ist „Nur IAM-Richtlinien“, wodurch Benutzern nur die Berechtigungen gewährt werden, die in ihrer IAM-Richtlinie enthalten sind.

- Wenn Sie einen Benutzer aus IAM oder aus einer anderen Region importieren, wird der Benutzer der Liste für alle vorhandenen Stacks mit der Berechtigungsstufe „Nur IAM-Richtlinien“ hinzugefügt.
- Standardmäßig hat ein Benutzer, den Sie gerade aus einer anderen Region importiert haben, keinen Zugriff auf Stacks in der Zielregion. Wenn Sie Benutzer aus einer anderen Region importieren, damit sie Stacks in der Zielregion verwalten können, müssen ihnen nach dem Import der Benutzer Berechtigungen für diese Stacks zugewiesen werden.
- Wenn Sie einen neuen Stack erstellen, werden alle aktuellen Benutzer automatisch der Liste mit der Berechtigungsebene IAM Policies Only (Nur IAM-Richtlinien) hinzugefügt.

Themen

Festlegen von Benutzerberechtigungen

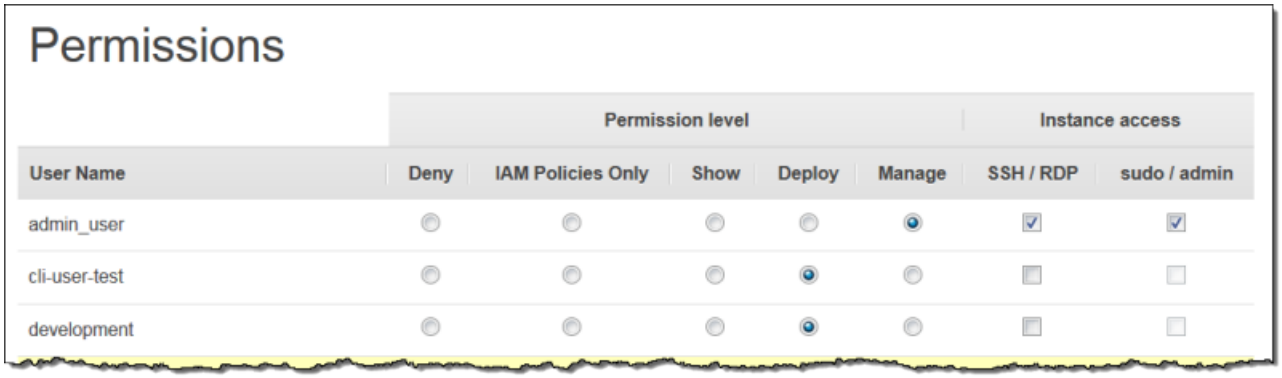
So legen Sie Benutzerberechtigungen fest

1. Wählen Sie im Navigationsbereich Permissions (Berechtigungen) aus.
2. Wählen Sie auf der Seite Permissions (Berechtigungen) die Option Edit (Bearbeiten) aus.
3. Ändern Sie die Einstellungen Permission level (Berechtigungsebene) und Instance access (Instance-Zugriff):

    - Weisen Sie über die Einstellung Permissions level (Berechtigungsebene) jedem Benutzer eine Standardberechtigungsebene zu, um festzulegen, ob der Benutzer auf diesen Stack zugreifen kann und welche Aktionen er dort ausführen darf. Wenn ein Benutzer über eine IAM-Richtlinie verfügt, bewertet AWS OpsWorks Stacks beide Berechtigungssätze. Ein Beispiel finden Sie unter Beispielrichtlinien.

- Über die Einstellung Instance access (Instance-Zugriff) SSH/RDP legen Sie fest, ob der Benutzer SSH-Zugriff (Linux) bzw. RDP-Zugriff (Windows) auf die Instances des Stacks hat.

  Wenn Sie SSH/RDP-Zugriff gewähren, können Sie optional auch sudo/admin auswählen, um dem Benutzer sudo-Berechtigungen (Linux) bzw. Administratorberechtigungen (Windows) für die Instances des Stacks zu gewähren.



Sie können Benutzer den folgenden Berechtigungsebenen zuweisen. Eine Liste der für jede Ebene zulässigen Aktionen finden Sie unter AWS OpsWorks Stapelt die Berechtigungsstufen.

Deny (Verweigern)

Der Benutzer kann keine AWS OpsWorks Stacks-Aktionen auf dem Stack ausführen, auch wenn er über eine IAM-Richtlinie verfügt, die AWS OpsWorks Stacks volle Zugriffsberechtigungen gewährt. Sie können diese Option beispielsweise nutzen, um Benutzern Zugriff auf Stacks für nicht freigegebene Produkte zu verweigern.

IAM Policies Only (Nur IAM-Richtlinien)

Dies ist die Standardebene, die allen neu importierten Benutzern und allen Benutzern für neu erstellte Stacks zugewiesen wird. Die Berechtigungen des Benutzers werden durch seine IAM-Richtlinie bestimmt. Wenn ein Benutzer keine IAM-Richtlinie hat oder seine Richtlinie keine expliziten AWS OpsWorks Stacks-Berechtigungen hat, kann er nicht auf den Stack zugreifen. Administratorbenutzern wird diese Stufe in der Regel zugewiesen, da ihre IAM-Richtlinien bereits vollständige Zugriffsberechtigungen gewähren.

Show (Anzeigen)

Der Benutzer kann einen Stack betrachten, aber keine Operationen darauf ausführen. Dies kann beispielsweise für Manager hilfreich sein, die die Stacks eines Kontos überwachen möchten, aber weder Apps bereitstellen noch den Stack bearbeiten müssen.

Bereitstellen

Diese Berechtigungsebene beinhaltet die Berechtigungsebene Show (Anzeigen) und ermöglicht es Benutzern darüber hinaus, Apps bereitzustellen. App-Entwickler müssen beispielsweise Updates auf den Instances eines Stacks bereitstellen, sie müssen dem Stack aber weder Layers noch Instances hinzufügen.

Verwalten

Diese Berechtigungsebene beinhaltet die Berechtigungsebene Deploy (Bereitstellen) und ermöglicht es Benutzern darüber hinaus, verschiedene Stack-Verwaltungsaufgaben auszuführen, darunter:

- Hinzufügen oder Löschen von Layers und Instances
- Zuweisen von Berechtigungsebenen auf der Seite Permissions (Berechtigungen) des Stacks.
- Registrieren oder Abmelden von Ressourcen

Jedem Stack kann beispielsweise ein eigener Verwalter zugewiesen sein, der sicherstellt, dass der Stack über eine ausreichende Anzahl und die richtigen Instances verfügt, der Paket- und Betriebssystemaktualisierungen verwaltet usw.

> **Note**
>
> Die Berechtigungsebene "Manage" verleiht Benutzern nicht die Berechtigung zum Erstellen oder Klonen von Stacks. Diese Berechtigungen müssen durch eine IAM-Richtlinie gewährt werden. Ein Beispiel finden Sie unter Verwalten von Berechtigungen.

Wenn der Benutzer auch über eine IAM-Richtlinie verfügt, bewertet AWS OpsWorks Stacks beide Berechtigungssätze. Auf diese Weise können Sie einem Benutzer eine Berechtigungsstufe zuweisen und anschließend eine Richtlinie anwenden, um die zulässigen Aktionen der Stufe einzuschränken oder zu erweitern. Sie könnten beispielsweise eine Richtlinie anwenden, die es einem Manage-Benutzer erlaubt, Stacks zu erstellen oder zu klonen, oder diesem Benutzer die Möglichkeit verweigert, Ressourcen zu registrieren oder zu deregistrieren. Weitere Beispiele für solche Richtlinien finden Sie unter Beispielrichtlinien.

> **Note**
>
> Wenn die Richtlinie des Benutzers zusätzliche Aktionen ermöglicht, kann es so wirken, als würden die Einstellungen der Seite Permissions (Berechtigungen) außer Kraft gesetzt. Wenn ein Benutzer beispielsweise über eine Richtlinie verfügt, die die CreateLayerAktion zulässt, Sie jedoch auf der Seite „Berechtigungen“ Bereitstellungsberechtigungen angeben, darf der Benutzer weiterhin Ebenen erstellen. Die Ausnahme von dieser Regel ist die Option Deny, mit der selbst Benutzern mit AWSOpsWorks_FullAccess Richtlinien der Zugriff auf den Stack verweigert wird. Weitere Informationen finden Sie unter Steuern des Zugriffs auf AWS Ressourcen mithilfe von Richtlinien.

Anzeigen der Berechtigungen

Wenn Sie die Selbstverwaltung aktiviert haben, können Benutzer über die Option My Settings (Eigene Einstellungen) oben rechts eine Zusammenfassung ihrer Berechtigungsebenen für die einzelnen Stacks anzeigen. Benutzer können auch auf Meine Einstellungen zugreifen, wenn ihre Richtlinie Berechtigungen für die UpdateMyUserProfileAktionen DescribeMyUserProfileund gewährt.

Verwenden von IAM-Bedingungsschlüsseln zur Überprüfung temporärer Anmeldeinformationen

AWS OpsWorks Stacks verfügt über eine integrierte Autorisierungsebene, die zusätzliche Autorisierungsfälle unterstützt (z. B. die vereinfachte Verwaltung des Nur-Lese- oder Lese-Schreibzugriffs auf Stacks für einzelne Benutzer). Dieser Autorisierungs-Layer ist von der Nutzung temporärer Anmeldeinformationen abhängig. Aus diesem Grund können Sie keine `aws:TokenIssueTime` Bedingung verwenden, um zu überprüfen, ob Benutzer langfristige Anmeldeinformationen verwenden, oder Aktionen von Benutzern blockieren, die temporäre Anmeldeinformationen verwenden, wie in der Referenz zu den IAM-JSON-Richtlinienelementen in der IAM-Dokumentation beschrieben.

Verwaltung von AWS OpsWorks Stacks-Berechtigungen durch Anhängen einer IAM-Richtlinie

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können die AWS OpsWorks Stacks-Berechtigungen eines Benutzers angeben, indem Sie eine IAM-Richtlinie anhängen. Bestimmte Berechtigungen können nur über eine angefügte Richtlinie verliehen werden:

- Berechtigungen für administrative Benutzer wie das Importieren von Benutzern
- Berechtigungen für bestimmte Aktionen wie das Erstellen und Klonen von Stacks

Eine vollständige Liste aller Aktionen, die nur mit einer angefügten Richtlinie möglich sind, finden Sie unter AWS OpsWorks Stapelt die Berechtigungsstufen.

Sie können eine Richtlinie auch verwenden, um die Berechtigungsstufen anzupassen, die über die Seite „Berechtigungen“ gewährt wurden. Dieser Abschnitt enthält eine kurze Zusammenfassung darüber, wie eine IAM-Richtlinie auf einen Benutzer angewendet wird, um AWS OpsWorks Stacks-Berechtigungen festzulegen. Weitere Informationen finden Sie unter Zugriffsverwaltung für AWS Ressourcen.

Eine IAM-Richtlinie ist ein JSON-Objekt, das eine oder mehrere Anweisungen enthält. Jedes Anweisungselement enthält eine Liste von Berechtigungen mit jeweils drei Grundelementen:

Action (Aktion)

Die Aktionen, auf die sich die Berechtigung auswirkt. Sie geben AWS OpsWorks Stacks-Aktionen als an. `opsworks:action` Eine `Action` kann eine bestimmte Aktion sein, beispielsweise `opsworks:CreateStack`, um festzulegen, ob ein Benutzer die Aktion `CreateStack` ausführen darf. Mithilfe von Platzhaltern können Sie auch Gruppen von Aktionen angeben. `opsworks:Create*` umfasst alle Aktionen zum Erstellen von Elementen. Eine vollständige Liste der AWS OpsWorks Stacks-Aktionen finden Sie in der AWS OpsWorks Stacks-API-Referenz.

Effect (Effekt)

Hierüber wird festgelegt, ob die angegebenen Aktionen erlaubt oder gesperrt sind.

Ressource

Die AWS Ressourcen, auf die sich die Genehmigung auswirkt. AWS OpsWorks Stacks hat einen Ressourcentyp, den Stack. Um Berechtigungen für eine bestimmte Stack-Ressource

zu verleihen, wählen Sie für `Resource` den ARN des Stacks im folgenden Format aus: `arn:aws:opsworks:`*`region`*`:`*`account_id`*`:stack/`*`stack_id`*`/`.

Sie können auch Platzhalter verwenden. Wenn Sie für `Resource` `*` auswählen, verleihen Sie Berechtigungen für alle Ressourcen.

Über die folgende Richtlinie wird dem Benutzer beispielsweise die Möglichkeit entzogen, Instances auf einem Stack mit der ID `2860-2f18b4cb-4de5-4429-a149-ff7da9f0d8ee` anzuhalten.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "opsworks:StopInstance",
      "Effect": "Deny",
      "Resource": "arn:aws:opsworks:*:*:stack/2f18b4cb-4de5-4429-a149-ff7da9f0d8ee/"
    }
  ]
}
```

Informationen zum Hinzufügen von Berechtigungen für einen IAM-Benutzer finden Sie unter. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_change-permissions.html#users_change_permissions-add-console

Weitere Informationen zum Erstellen oder Ändern von IAM-Richtlinien finden Sie unter Richtlinien und Berechtigungen in IAM. Einige Beispiele für AWS OpsWorks Stacks-Richtlinien finden Sie unter. Beispielrichtlinien

Beispielrichtlinien

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Abschnitt werden Beispiele für IAM-Richtlinien beschrieben, die auf Stacks-Benutzer angewendet werden können. AWS OpsWorks

- Administrative Berechtigungenbeschreibt Richtlinien, die verwendet werden, um Administratorbenutzern Berechtigungen zu gewähren.
- Verwalten von Berechtigungenund "Deploy"-Berechtigungen zeigt Beispiele für Richtlinien, die auf einen Benutzer angewendet werden können, um die Berechtigungsstufen „Verwalten“ und „Bereitstellen“ zu erweitern oder einzuschränken.

  AWS OpsWorks Stacks ermittelt die Berechtigungen des Benutzers, indem es die durch IAM-Richtlinien erteilten Berechtigungen sowie die auf der Seite „Berechtigungen“ gewährten Berechtigungen auswertet. Weitere Informationen finden Sie unter Steuern des Zugriffs auf AWS Ressourcen mithilfe von Richtlinien. Weitere Informationen zu den Berechtigungen auf der Seite Permissions (Berechtigungen) finden Sie unter AWS OpsWorks Stapelt die Berechtigungsstufen.

Administrative Berechtigungen

Verwenden Sie die IAM-Konsole, https://console.aws.amazon.com/iam/, um auf die AWSOpsWorks_FullAccess Richtlinie zuzugreifen. Hängen Sie diese Richtlinie einem Benutzer an, um ihm Berechtigungen zur Ausführung aller AWS OpsWorks Stacks-Aktionen zu gewähren. Die IAM-Berechtigungen sind unter anderem erforderlich, damit ein Administratorbenutzer Benutzer importieren kann.

Sie müssen eine IAM-Rolle erstellen, die es AWS OpsWorks Stacks ermöglicht, in Ihrem Namen auf andere AWS Ressourcen wie Amazon EC2 EC2-Instances zuzugreifen. In der Regel erledigen Sie diese Aufgabe, indem Sie einen Administratorbenutzer den ersten Stack erstellen lassen und AWS OpsWorks Stacks die Rolle für Sie erstellen lassen. Diese Rolle können Sie dann für alle weiteren Stacks verwenden. Weitere Informationen finden Sie unter AWS OpsWorks Stacks erlauben, in Ihrem Namen zu handeln.

Der Administratorbenutzer, der den ersten Stack erstellt, muss über Berechtigungen für einige IAM-Aktionen verfügen, die nicht in der AWSOpsWorks_FullAccess Richtlinie enthalten sind. Fügen Sie dem `Actions` Abschnitt der Richtlinie die folgenden Berechtigungen hinzu. Achten Sie für eine korrekte JSON-Syntax darauf, Kommas zwischen den Aktionen hinzuzufügen und das nachstehende Komma am Ende der Aktionsliste zu entfernen.

```
"iam:PutRolePolicy",
"iam:AddRoleToInstanceProfile",
"iam:CreateInstanceProfile",
```

```
"iam:CreateRole"
```

Verwalten von Berechtigungen

Mit der Berechtigungsebene Manage (Verwalten) können Benutzer eine Reihe von Stackverwaltungsaufgaben wie das Anlegen oder Löschen von Layers ausführen. In diesem Thema werden mehrere Richtlinien beschrieben, mit denen Sie Benutzer verwalten können, um die Standardberechtigungen zu erweitern oder einzuschränken.

Entziehen Sie einem Benutzer mit der Berechtigungsebene Manage (Verwalten) die Möglichkeit, Layers anzulegen oder zu löschen.

Mithilfe der folgenden IAM-Richtlinie können Sie die Berechtigungsstufe „Verwalten“ einschränken, sodass ein Benutzer alle Verwaltungsaktionen ausführen kann, mit Ausnahme des Hinzufügens oder Löschens von Ebenen. Ersetzen Sie *region*, *account_id* und *stack_id* durch Werte, die Ihrer Konfiguration entsprechen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "opsworks:CreateLayer",
        "opsworks:DeleteLayer"
      ],
      "Resource": "arn:aws:opsworks:region:account_id:stack/stack_id/"
    }
  ]
}
```

Erlauben Sie einem Benutzer mit der Berechtigungsebene Manage (Verwalten) das Erstellen und Klonen von Stacks.

Die Berechtigungsstufe „Verwalten“ erlaubt es Benutzern nicht, Stacks zu erstellen oder zu klonen. Sie können die Verwaltungsberechtigungen so ändern, dass ein Benutzer Stacks erstellen oder klonen kann, indem Sie die folgende IAM-Richtlinie anwenden. Ersetzen Sie *region* und *account_id* durch Werte, die Ihrer Konfiguration entsprechen.

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

Entziehen Sie einem Benutzer mit der Berechtigungsebene "Manage" die Möglichkeit, Ressourcen zu registrieren und abzumelden.

Die Stufe „Berechtigungen verwalten“ ermöglicht es dem Benutzer, Amazon EBS- und Elastic IP-Adressressourcen beim Stack zu registrieren und zu deregistrieren. Sie können die Verwaltungsberechtigungen einschränken, sodass der Benutzer alle Verwaltungsaktionen mit Ausnahme der Registrierung von Ressourcen ausführen kann, indem Sie die folgende Richtlinie anwenden.

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

```
      "Action": [
        "opsworks:RegisterVolume",
        "opsworks:RegisterElasticIp"
      ],
      "Resource": "*"
    }
  ]
}
```

Verleihen Sie einem Benutzer mit der Berechtigungsebene Manage (Verwalten) die Berechtigung, Benutzer zu importieren.

Die Berechtigungsstufe „Verwalten“ erlaubt es Benutzern nicht, Benutzer in AWS OpsWorks Stacks zu importieren. Sie können die Verwaltungsberechtigungen so erweitern, dass ein Benutzer Benutzer importieren und löschen kann, indem Sie die folgende IAM-Richtlinie anwenden. Ersetzen Sie *region* und *account_id* durch Werte, die Ihrer Konfiguration entsprechen.

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

"Deploy"-Berechtigungen

Benutzer mit der Berechtigungsebene Deploy (Bereitstellen) können keine Apps erstellen oder löschen. Sie können die Bereitstellungsberechtigungen erweitern, sodass ein Benutzer Apps erstellen und löschen kann, indem Sie die folgende IAM-Richtlinie anwenden. Ersetzen Sie *region*, *account_id und stack_id* durch Werte, die Ihrer Konfiguration entsprechen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "opsworks:CreateApp",
        "opsworks:DeleteApp"
      ],
      "Resource": "arn:aws:opsworks:region:account_id:stack/stack_id/"
    }
  ]
}
```

AWS OpsWorks Stapelt die Berechtigungsstufen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Abschnitt sind die Aktionen aufgeführt, die durch die Berechtigungsstufen Anzeigen, Bereitstellen und Verwalten auf der Seite AWS OpsWorks Stacks-Berechtigungen zulässig sind. Er enthält auch eine Liste von Aktionen, denen Sie Berechtigungen nur gewähren können, indem Sie dem Benutzer eine IAM-Richtlinie zuweisen.

Show (Anzeigen)

Mit der Berechtigungsebene Show (Anzeigen) können Sie `Describe`*XYZ*-Befehle mit folgenden Ausnahmen ausführen:

```
DescribePermissions
DescribeUserProfiles
DescribeMyUserProfile
DescribeStackProvisioningParameters
```

Wenn ein administrativer Benutzer die Selbstverwaltung für einen Benutzer aktiviert hat, können Benutzer mit der Berechtigungsebene Show (Anzeigen) auch die Optionen `DescribeMyUserProfile` und `UpdateMyUserProfile` nutzen. Weitere Informationen zur Selbstverwaltung finden Sie unter Bearbeiten von Benutzereinstellungen.

Bereitstellen

Folgende Aktionen sind mit der Berechtigungsebene Deploy (Bereitstellen) zusätzlich zu den Aktionen der Berechtigungsebene Show (Anzeigen) erlaubt.

```
CreateDeployment
UpdateApp
```

Verwalten

Folgende Aktionen sind mit der Berechtigungsebene Manage (Verwalten) zusätzlich zu den Aktionen der Berechtigungsebenen Deploy (Bereitstellen) und Show (Anzeigen) erlaubt.

```
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
```

Berechtigungen, für die eine IAM-Richtlinie erforderlich ist

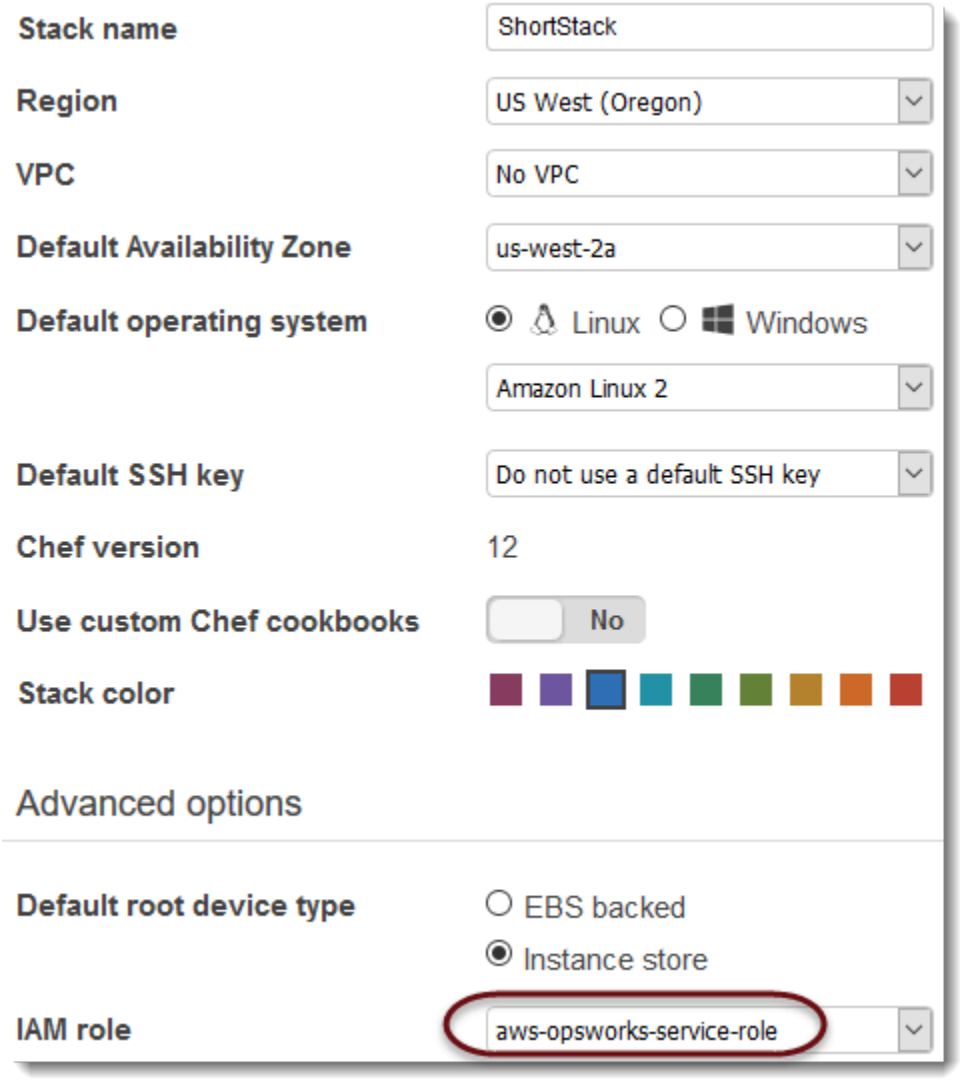
Sie müssen dem Benutzer Berechtigungen für die folgenden Aktionen gewähren, indem Sie eine entsprechende IAM-Richtlinie anwenden. Einige Beispiele finden Sie unter Beispielrichtlinien.

```
CloneStack
CreateStack
CreateUserProfile
DeleteUserProfile
DescribeUserProfiles
UpdateUserProfile
```

AWS OpsWorks Stacks erlauben, in Ihrem Namen zu handeln

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks muss in Ihrem Namen mit einer Vielzahl von AWS-Services interagieren. AWS OpsWorks Stacks interagiert beispielsweise mit Amazon EC2, um Instances zu erstellen, und mit Amazon, CloudWatch um Überwachungsstatistiken abzurufen. Wenn Sie einen Stack erstellen, geben Sie eine IAM-Rolle an, die normalerweise als Servicerolle bezeichnet wird und AWS OpsWorks Stacks die entsprechenden Berechtigungen gewährt.



Wenn Sie eine neue Stack-Servicerolle angeben, können Sie einen der folgenden Schritte ausführen:

- Geben Sie eine Standard-Servicerolle an, die Sie zuvor erstellt haben.

  Sie können in der Regel beim Erstellen Ihres ersten Stacks eine Standard-Servicerolle erstellen und diese anschließend als Rolle für alle nachfolgenden Stacks verwenden.
- Geben Sie eine benutzerdefinierte Servicerolle an, die Sie mithilfe der IAM-Konsole oder API erstellt haben.

  Dieser Ansatz ist nützlich, wenn Sie AWS OpsWorks Stacks mehr eingeschränkte Berechtigungen als der Standard-Servicerolle gewähren möchten.

> **Note**
>
> Um Ihren ersten Stack zu erstellen, müssen Sie über die in der AdministratorAccessIAM-Richtlinienvorlage definierten Berechtigungen verfügen. Diese Berechtigungen erlauben AWS OpsWorks Stacks, eine neue IAM-Servicerolle anzulegen und Ihnen, Benutzer zu importieren, wie zuvor beschrieben. Für alle nachfolgenden Stacks können Benutzer die Servicerolle auswählen, die sie für den ersten Stack erstellt haben; sie brauchen keine vollständigen Administratorberechtigungen, um einen Stack zu erstellen.

Die Standard-Servicerolle gewährt folgende Berechtigungen:

- Führen Sie alle Amazon EC2 EC2-Aktionen aus (`ec2:*`).
- CloudWatch Statistiken abrufen (`cloudwatch:GetMetricStatistics`).
- Verwenden Sie Elastic Load Balancing, um den Traffic auf Server zu verteilen (`elasticloadbalancing:*`).
- Verwenden Sie eine Amazon RDS-Instance als Datenbankserver (`rds:*`).
- Verwenden Sie IAM-Rollen (`iam:PassRole`), um eine sichere Kommunikation zwischen AWS OpsWorks Stacks und Ihren Amazon EC2 EC2-Instances bereitzustellen.

Wenn Sie eine benutzerdefinierte Servicerolle erstellen, müssen Sie sicherstellen, dass sie alle Berechtigungen gewährt, die Stacks zur Verwaltung AWS OpsWorks Ihres Stacks benötigt. Das folgende JSON-Beispiel zeigt die Richtlinienanweisung für die Standard-Servicerolle. Eine benutzerdefinierte Service-Rolle muss mindestens die folgenden Berechtigungen in ihrer Richtlinienanweisung enthalten.

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

Ein Servicerolle hat zudem eine Vertrauensstellung. Von AWS OpsWorks Stacks erstellte Servicerollen haben die folgende Vertrauensstellung.

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

Die Servicerolle muss über dieses Vertrauensverhältnis verfügen, damit AWS OpsWorks Stacks in Ihrem Namen handeln kann. Ändern Sie die Vertrauensstellung nicht, wenn Sie die Standard-Servicerolle verwenden. Wenn Sie eine benutzerdefinierte Servicerolle erstellen, geben Sie die Vertrauensstellung an, indem Sie einen der folgenden Schritte ausführen:

- Wenn Sie den Assistenten zum Erstellen von Rollen in der IAM-Konsole verwenden, wählen Sie unter Anwendungsfall auswählen die Option Opsworks aus. Diese Rolle hat die entsprechende Vertrauensstellung, aber es ist nicht implizit eine Richtlinie beigefügt. Um AWS OpsWorks Stacks

die Erlaubnis zu erteilen, in Ihrem Namen zu handeln, erstellen Sie eine vom Kunden verwaltete Richtlinie, die Folgendes enthält, und fügen Sie sie der neuen Rolle hinzu.

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

- Wenn Sie eine AWS CloudFormation Vorlage verwenden, können Sie dem Abschnitt Ressourcen Ihrer Vorlage etwas wie das Folgende hinzufügen.

```
"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": {
              ...
            } ]
        }
      },
   }
}
```

Dienstübergreifende Verhinderung verwirrter Abgeordneter in AWS OpsWorks Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiertere Entität zur Durchführung der Aktion zwingen kann. In AWS kann ein dienstübergreifendes Identitätswechsels zu einem Problem mit dem verwirrten Stellvertreter führen. Ein dienstübergreifender Identitätswechsel kann auftreten,

wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen, die Kontextschlüssel `aws:SourceArn`und die `aws:SourceAccount`globalen Bedingungsschlüssel in den Richtlinien für den Zugriff auf Stacks zu verwenden, um die Berechtigungen zu beschränken, die Stacks einem anderen Dienst für AWS OpsWorks Stacks erteilt. Wenn der `aws:SourceArn`-Wert die Konto-ID nicht enthält, z. B. einen Amazon-S3-Bucket-ARN, müssen Sie beide globale Bedingungskontextschlüssel verwenden, um Berechtigungen einzuschränken. Wenn Sie beide globale Bedingungskontextschlüssel verwenden und der `aws:SourceArn`-Wert die Konto-ID enthält, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in der gleichen Richtlinienanweisung verwendet wird. Verwenden Sie diese `aws:SourceArn` Option, wenn Sie möchten, dass nur ein Stack mit dem dienstübergreifenden Zugriff verknüpft wird. Verwenden Sie diese Option, `aws:SourceAccount` wenn Sie zulassen möchten, dass ein beliebiger Stapel in diesem Konto mit der dienstübergreifenden Nutzung verknüpft wird.

Der Wert von `aws:SourceArn` muss der ARN eines AWS OpsWorks Stacks sein.

Der effektivste Weg, sich vor dem Problem des verwirrten Stellvertreters zu schützen, besteht darin, den Kontextschlüssel für `aws:SourceArn` globale Bedingungen mit dem vollständigen ARN des AWS OpsWorks Stacks-Stacks zu verwenden. Wenn Sie den vollständigen ARN nicht kennen oder wenn Sie mehrere Stack-ARNs angeben, verwenden Sie den `aws:SourceArn` globalen Kontextbedingungsschlüssel mit Platzhaltern (*) für die unbekannten Teile des ARN. z. B. `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Der folgende Abschnitt zeigt, wie Sie die Kontextschlüssel `aws:SourceArn` und die `aws:SourceAccount` globalen Bedingungsschlüssel in AWS OpsWorks Stacks verwenden können, um das Problem mit dem verwirrten Deputy zu vermeiden.

Beugen Sie Exploits mit verwirrten Stellvertretern in Stacks vor AWS OpsWorks

In diesem Abschnitt wird beschrieben, wie Sie dazu beitragen können, Exploits mit verwirrten Stellvertretern in AWS OpsWorks Stacks zu verhindern. Außerdem finden Sie Beispiele für Berechtigungsrichtlinien, die Sie an die IAM-Rolle anhängen können, die Sie für den Zugriff auf Stacks verwenden. AWS OpsWorks Aus Sicherheitsgründen empfehlen wir, die

Schlüssel `aws:SourceArn` und die `aws:SourceAccount` Bedingungsschlüssel zu den Vertrauensbeziehungen hinzuzufügen, die Ihre IAM-Rolle mit anderen Diensten unterhält. Die Vertrauensbeziehungen ermöglichen es AWS OpsWorks Stacks, eine Rolle bei der Ausführung von Aktionen in anderen Diensten zu übernehmen, die für die Erstellung oder Verwaltung Ihrer AWS OpsWorks Stacks-Stacks erforderlich sind.

Um Vertrauensstellungen zu bearbeiten, Schlüssel hinzuzufügen und zu konditionieren **`aws:SourceArnaws:SourceAccount`**

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. Suchen Sie im Suchfeld nach der Rolle, die Sie für den Zugriff auf AWS OpsWorks Stacks verwenden. Die AWS verwaltete Rolle ist`aws-opsworks-service-role`.
4. Wählen Sie auf der Übersichtsseite für die Rolle die Registerkarte Vertrauensbeziehungen aus.
5. Wählen Sie auf der Registerkarte Vertrauensbeziehungen die Option Vertrauensrichtlinie bearbeiten aus.
6. Fügen Sie auf der Seite Vertrauensrichtlinie bearbeiten der Richtlinie mindestens einen der `aws:SourceAccount` Bedingungsschlüssel `aws:SourceArn` oder einen der Bedingungsschlüssel hinzu. Wird verwendet`aws:SourceArn`, um die Vertrauensbeziehung zwischen Cross-Services (wie Amazon EC2) und AWS OpsWorks Stacks auf bestimmte AWS OpsWorks Stacks-Stacks zu beschränken, was restriktiver ist. Fügen Sie hinzu`aws:SourceAccount`, um die Vertrauensbeziehung zwischen Cross-Services und AWS OpsWorks Stacks auf Stacks in einem bestimmten Konto zu beschränken, was weniger restriktiv ist. Im Folgenden wird ein Beispiel gezeigt. Beachten Sie, dass die Konto-IDs identisch sein müssen, wenn Sie beide Bedingungsschlüssel verwenden.

```
{
  "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. Wenn Sie mit dem Hinzufügen von Bedingungsschlüsseln fertig sind, wählen Sie Richtlinie aktualisieren.

Im Folgenden finden Sie weitere Beispiele für Rollen, die den Zugriff auf Stacks mithilfe von `aws:SourceArn` und `aws:SourceAccount` einschränken.

Themen

- Beispiel: Zugriff auf Stacks in einer bestimmten Region
- Beispiel: Hinzufügen von mehr als einem Stack-ARN zu aws:SourceArn

Beispiel: Zugriff auf Stacks in einer bestimmten Region

Die folgende Erklärung zur Rollenvertrauensstellung greift auf alle AWS OpsWorks Stacks-Stacks in der Region USA Ost (Ohio) zu (). `us-east-2` Beachten Sie, dass die Region im ARN-Wert von angegeben ist`aws:SourceArn`, der Stack-ID-Wert jedoch ein Platzhalter (*) ist.

```
{
  "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/*"
```

```
        }
      }
    }
  ]
}
```

Beispiel: Hinzufügen von mehr als einem Stack-ARN zu **`aws:SourceArn`**

Das folgende Beispiel beschränkt den Zugriff auf ein Array von zwei AWS OpsWorks Stacks-Stacks in der Konto-ID 123456789012.

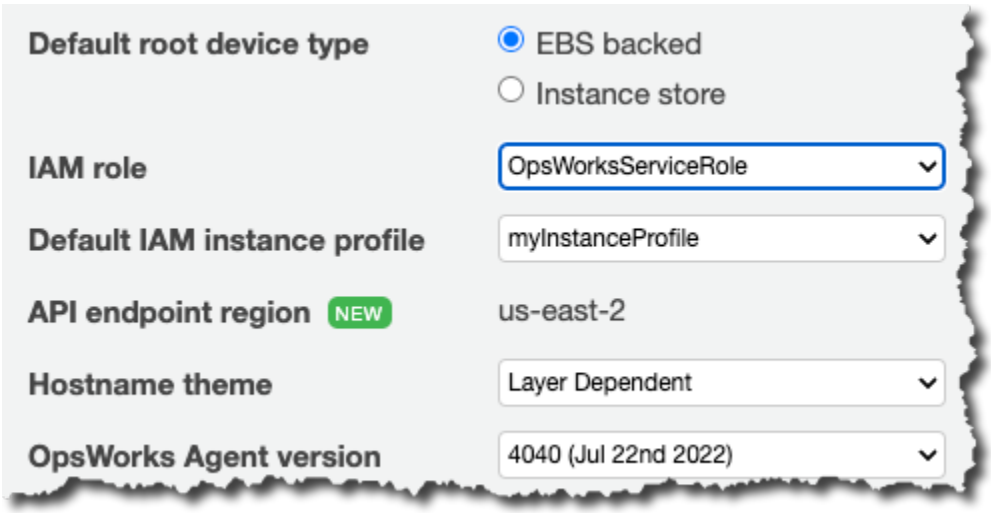
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "opsworks.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnEquals": {
          "aws:SourceArn": [
             "arn:aws:opsworks:us-east-2:123456789012:stack/unique_ID1",
             "arn:aws:opsworks:us-east-2:123456789012:stack/unique_ID2"
           ]
       }
      }
    }
  ]
}
```

Festlegen von Berechtigungen für Apps auf EC2-Instances

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn die Anwendungen, die auf den Amazon EC2 EC2-Instances Ihres Stacks ausgeführt werden, auf andere AWS-Ressourcen wie Amazon S3 S3-Buckets zugreifen müssen, müssen sie über die entsprechenden Berechtigungen verfügen. Diese Berechtigungen können Sie über ein Instance-Profil gewähren. Sie können für jede Instance ein Instance-Profil angeben, wenn Sie einen AWS OpsWorks Stacks-Stack erstellen.



Sie können durch Bearbeiten der Layer-Konfiguration auch ein Profil für die Instances eines Layers festlegen.

Über das Instance-Profil wird eine IAM-Rolle festgelegt. Anwendungen auf einer Instance können mithilfe dieser Rolle auf AWS-Ressourcen zugreifen und verwenden dafür die Berechtigungen, die über die Richtlinie der Rolle gewährt werden. Weitere Informationen dazu, wie Anwendungen Rollen übernehmen, finden Sie unter Assuming the Role Using an API Call (Übernehmen der Rolle mit einem API-Aufruf).

Sie haben folgende Möglichkeiten, um ein Instance-Profil zu erstellen:

- Verwenden Sie die IAM-Konsole oder API, um ein Profil zu erstellen.

  Weitere Informationen finden Sie unter Rollen (Übertragung und Vereinigung).
- Verwenden Sie eine AWS CloudFormation Vorlage, um ein Profil zu erstellen.

  Einige Beispiele dafür, wie Sie IAM-Ressourcen in eine Vorlage aufnehmen können, finden Sie unter Vorlagenausschnitte für Identity and Access Management (IAM).

Ein Instance-Profil benötigt eine Vertrauensbeziehung und eine angefügte Richtlinie, die die erforderlichen Berechtigungen zum Zugriff auf AWS-Ressourcen verleiht.

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

Das Instanzprofil muss über diese Vertrauensbeziehung verfügen, damit AWS OpsWorks Stacks in Ihrem Namen handeln kann. Ändern Sie die Vertrauensstellung nicht, wenn Sie die Standard-Servicerolle verwenden. Wenn Sie eine benutzerdefinierte Servicerolle erstellen, geben Sie die Vertrauensstellung wie folgt an:

- Wenn Sie den Assistenten Create Role (Rolle erstellen) in der IAM-Konsole verwenden, geben Sie den Rollentyp Amazon EC2 unter AWS Service Roles (AWS-Servicerolle) auf der zweiten Seite des Assistenten an.
- Wenn Sie eine AWS CloudFormation Vorlage verwenden, können Sie dem Abschnitt Ressourcen Ihrer Vorlage etwas wie das Folgende hinzufügen.

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

Wenn Sie Ihr Instanzprofil erstellen, können Sie der jeweiligen Rolle des Profils eine entsprechende Richtlinie zuordnen. Nachdem Sie den Stack erstellt haben, müssen Sie die IAM-Konsole oder API verwenden, um der Rolle des Profils eine entsprechende Richtlinie zuzuweisen. Die folgende Richtlinie gewährt beispielsweise vollen Zugriff auf alle Objekte im Amazon S3 S3-Bucket mit dem Namen DOC-EXAMPLE-BUCKET. Ersetzen Sie *region* und DOC-EXAMPLE-BUCKET durch Werte, die Ihrer Konfiguration entsprechen.

```
{
  "Version": "2012-10-17",
  "Statement": [ {
    "Effect": "Allow",
    "Action": "s3:*",
    "Resource": "arn:aws:s3:region::DOC-EXAMPLE-BUCKET/*"
    }
  ]
}
```

Ein Beispiel für das Erstellen und Verwenden von Instance-Profilen finden Sie unter Verwenden eines Amazon S3-Buckets.

Wenn Ihre Anwendung ein Instance-Profil verwendet, um die AWS OpsWorks Stacks-API von einer EC2-Instance aus aufzurufen, muss die Richtlinie die `iam:PassRole` Aktion zusätzlich zu den entsprechenden Aktionen für AWS OpsWorks Stacks und andere AWS-Services zulassen. Über die Berechtigung `iam:PassRole` kann AWS OpsWorks Stacks die Service-Rolle in Ihrem Namen

übernehmen. Weitere Informationen zur AWS OpsWorks Stacks-API finden Sie unter OpsWorks AWS-API-Referenz.

Im Folgenden finden Sie ein Beispiel für eine IAM-Richtlinie, mit der Sie jede AWS OpsWorks Stacks-Aktion von einer EC2-Instance sowie jede Amazon EC2- oder Amazon S3 S3-Aktion aufrufen können.

```
{
  "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

Wenn Sie dies nicht zulassen`iam:PassRole`, schlägt jeder Versuch, eine AWS OpsWorks Stacks-Aktion aufzurufen, mit einem Fehler wie dem folgenden fehl:

```
User: arn:aws:sts::123456789012:federated-user/Bob is not authorized
to perform: iam:PassRole on resource:
arn:aws:sts::123456789012:role/OpsWorksStackIamRole
```

Weitere Informationen dazu, wie Sie mithilfe von Rollen Berechtigungen auf EC2-Instances verleihen, finden Sie unter Gewähren von Zugriff auf AWS-Ressourcen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden im AWS Identity and Access Management -Benutzerhandbuch.

Verwalten des SSH-Zugriffs

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks unterstützt SSH-Schlüssel sowohl für Linux- als auch für Windows-Stacks.

- Für Linux-Instances können Sie SSH zum Anmelden bei einer Instance verwenden, um z. B. Agenten-CLI-Befehle auszuführen.

  Weitere Informationen finden Sie unter Anmelden mit SSH.
- Für Windows-Instances können Sie einen SSH-Schlüssel zum Abrufen des Administratorpassworts der Instance verwenden, mit dem Sie sich dann bei RDP anmelden können.

  Weitere Informationen finden Sie unter Anmelden mit RDP.

Die Authentifizierung basiert auf einem SSH-Schlüsselpaar, das aus einem öffentlichen und einem privaten Schlüssel besteht:

- Installieren Sie den öffentlichen Schlüssel auf der Instance.

  Der Speicherort hängt vom jeweiligen Betriebssystem ab, aber AWS OpsWorks Stacks kümmert sich um die Details für Sie.
- Speichern Sie den privaten Schlüssel lokal und stellen Sie ihn zum Zugriff auf die Instance für einen SSH-Client bereit, z. B. `ssh.exe`.

  Der SSH-Client verwendet den privaten Schlüssel, um eine Verbindung mit der Instance herzustellen.

Um den Benutzern eines Stacks SSH-Zugriff zu erteilen, müssen Sie die SSH-Schlüsselpaare erstellen, die öffentlichen Schlüssel auf den Stack-Instances installieren und die privaten Schlüssel sicher verwalten.

Amazon EC2 bietet eine einfache Möglichkeit, einen öffentlichen SSH-Schlüssel auf einer Instance zu installieren. Sie können die Amazon EC2 EC2-Konsole oder API verwenden, um ein oder mehrere Schlüsselpaare für jede AWS-Region zu erstellen, die Sie verwenden möchten. Amazon EC2 speichert die öffentlichen Schlüssel auf AWS und Sie speichern die privaten Schlüssel lokal. Wenn Sie eine Instance starten, geben Sie eines der Schlüsselpaare der Region an und Amazon EC2 installiert es automatisch auf der Instance. Anschließend verwenden Sie den entsprechenden privaten Schlüssel zum Anmelden bei der Instance. Weitere Informationen finden Sie unter Amazon EC2-Schlüsselpaare.

Mit AWS OpsWorks Stacks können Sie eines der Amazon EC2 EC2-Schlüsselpaare der Region angeben, wenn Sie einen Stack erstellen, und es optional mit einem anderen key pair überschreiben, wenn Sie jede Instance erstellen. Wenn AWS OpsWorks Stacks die entsprechende Amazon EC2-Instance startet, gibt es das key pair an und Amazon EC2 installiert den öffentlichen Schlüssel auf der Instance. Sie können dann den privaten Schlüssel verwenden, um sich anzumelden oder ein Administratorkennwort abzurufen, genau wie bei einer standardmäßigen Amazon EC2 EC2-Instance. Weitere Informationen finden Sie unter Installation eines Amazon EC2 EC2-Schlüssels.

Die Verwendung eines Amazon EC2 EC2-Schlüsselpaars ist praktisch, hat jedoch zwei wesentliche Einschränkungen:

- Ein Amazon EC2 EC2-Schlüsselpaar ist an eine bestimmte AWS-Region gebunden.

  Wenn Sie in mehreren Regionen arbeiten, müssen Sie mehrere Schlüsselpaare verwalten.
- Sie können nur ein Amazon EC2 EC2-Schlüsselpaar auf einer Instance installieren.

  Wenn Sie mehreren Benutzern die Anmeldung ermöglichen möchten, müssen sie alle eine Kopie des privaten Schlüssels haben. Aus Sicherheitsgründen empfiehlt sich dies jedoch nicht.

Für Linux-Stacks bietet AWS OpsWorks Stacks eine einfachere und flexiblere Möglichkeit, SSH-Schlüsselpaare zu verwalten.

- Jeder Benutzer registriert ein persönliches Schlüsselpaar.

Sie speichern den privaten Schlüssel lokal und registrieren den öffentlichen Schlüssel bei AWS OpsWorks Stacks, wie unter beschrieben. Registrierung des öffentlichen SSH-Schlüssels eines Benutzers

- Wenn Sie Benutzerberechtigungen für einen Stack festlegen, geben Sie an, welche Benutzer SSH-Zugriff auf die Stack-Instances haben sollen.

  AWS OpsWorks Stacks erstellt automatisch für jeden autorisierten Benutzer einen Systembenutzer auf den Instanzen des Stacks und installiert seinen öffentlichen Schlüssel. Der Benutzer kann sich dann mit dem entsprechenden privaten Schlüssel anmelden, wie unter Anmelden mit SSH beschrieben.

Die Verwendung persönlicher SSH-Schlüssel bietet folgende Vorteile.

- Es ist nicht erforderlich, Schlüssel auf den Instances manuell zu konfigurieren. AWS OpsWorks Stacks installiert automatisch die entsprechenden öffentlichen Schlüssel auf jeder Instanz.
- AWS OpsWorks Stacks installiert nur die persönlichen öffentlichen Schlüssel autorisierter Benutzer.

  Nicht autorisierte Benutzer können nicht mit ihrem persönlichen privaten Schlüssel auf Instances zugreifen. Mit Amazon EC2 EC2-Schlüsselpaaren kann sich jeder Benutzer mit dem entsprechenden privaten Schlüssel anmelden, mit oder ohne autorisierten SSH-Zugriff.
- Wenn ein Benutzer den SSH-Zugriff nicht mehr benötigt, können Sie auf der Seite Berechtigungen die SSH/RDP-Berechtigungen des Benutzers aufheben.

  AWS OpsWorks Stacks deinstalliert sofort den öffentlichen Schlüssel aus den Instances des Stacks.
- Sie können denselben Schlüssel für jede AWS-Region verwenden.

  Benutzer müssen nur einen privaten Schlüssel verwalten.
- Es ist nicht erforderlich, private Schlüssel freizugeben.

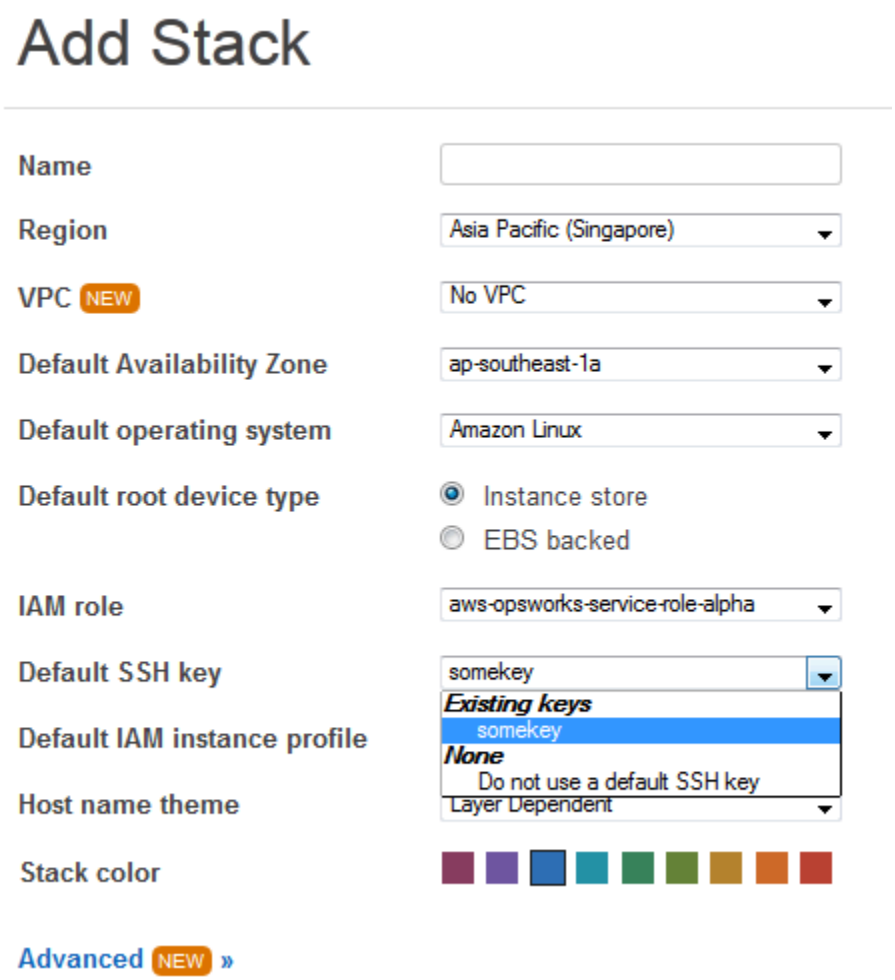
  Jeder Benutzer hat seinen eigenen privaten Schlüssel.
- Die wechselseitige Verwendung von Schlüsseln ist ganz einfach.

  Sie oder der Benutzer aktualisiert den öffentlichen Schlüssel unter My Settings (Eigene Einstellungen) und AWS OpsWorks Stacks aktualisiert die Instances automatisch.

Installation eines Amazon EC2 EC2-Schlüssels

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

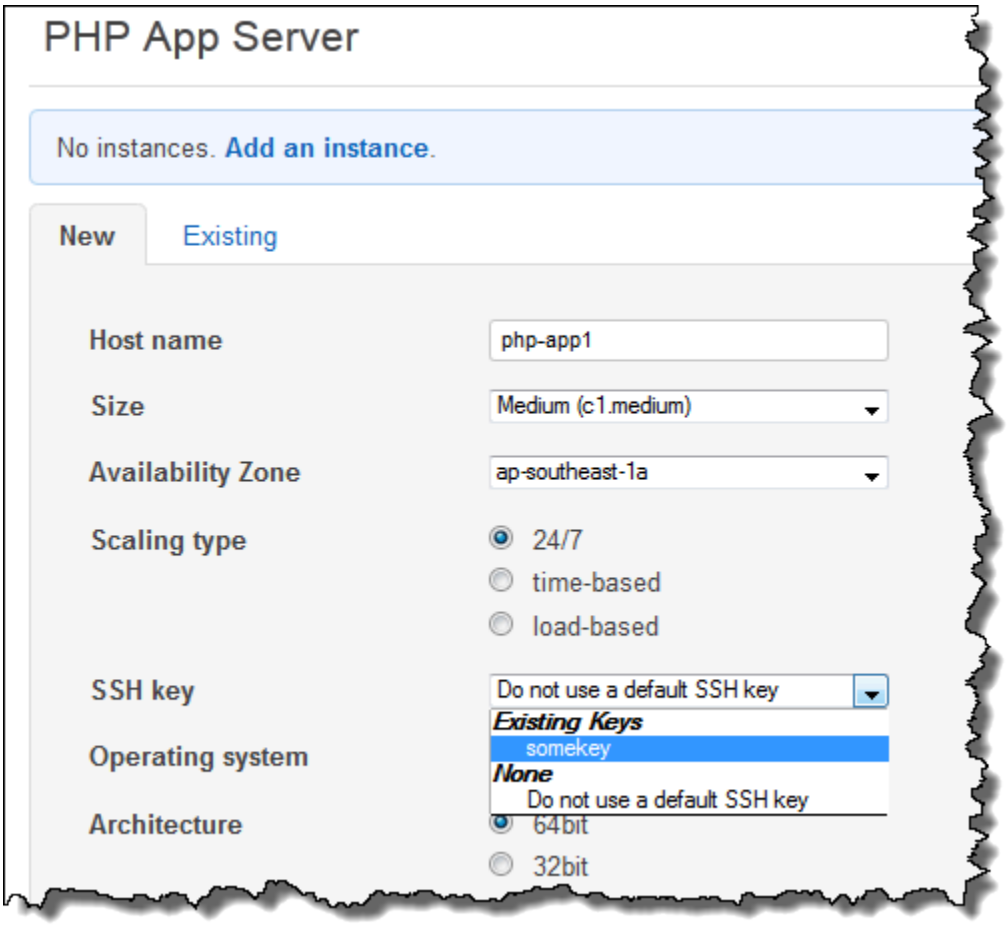
Wenn Sie einen Stack erstellen, können Sie einen Amazon EC2 EC2-SSH-Schlüssel angeben, der standardmäßig auf jeder Instance im Stack installiert ist.



Die Standard-SSH-Schlüsselliste zeigt die Amazon EC2-Schlüssel Ihres AWS-Kontos. Sie können einen der folgenden Schritte ausführen:

- Wählen Sie den entsprechenden Schlüssel in der Liste aus.
- Wählen Sie Do not use a default SSH key (Keinen Standard-SSH-Schlüssel verwenden) aus, um keinen Schlüssel anzugeben.

Wenn Sie Do not use a default SSH key (Keinen Standard-SSH-Schlüssel verwenden) ausgewählt haben oder den Standardschlüssel eines Stacks überschreiben möchten, können Sie beim Erstellen einer Instance einen Schlüssel festlegen.



Wenn Sie die Instance starten, installiert AWS OpsWorks Stacks den öffentlichen Schlüssel in der Datei. `authorized_keys`

Registrierung des öffentlichen SSH-Schlüssels eines Benutzers

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir

empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Es gibt zwei Methoden zum Registrieren des öffentlichen SSH-Schlüssels eines Benutzers:

- Ein Benutzer mit Administratorrechten kann einem oder mehreren Benutzern einen öffentlichen SSH-Schlüssel zuweisen und ihnen den entsprechenden privaten Schlüssel bereitstellen.
- Ein Benutzer mit Administratorrechten kann für einen oder mehrere Benutzer die Funktion zur Selbstverwaltung aktivieren.

  Diese Benutzer können dann ihren eigenen öffentlichen SSH-Schlüssel festlegen.

Weitere Informationen dazu, wie Benutzer mit Administratorrechten die Selbstverwaltung aktivieren oder Benutzern öffentliche Schlüssel zuweisen können, finden Sie unter Bearbeiten von Benutzereinstellungen.

Für die Herstellung einer Verbindung mit Linux-basierten Instances in einem PuTTY-Terminal mithilfe von SSH sind zusätzliche Schritte erforderlich. Weitere Informationen finden Sie in der AWS-Dokumentation unter Herstellung einer Verbindung zu Ihrer Linux-Instance von Windows mit PuTTY und Beheben von Verbindungsproblemen mit Ihrer Instance.

Im Folgenden wird beschrieben, wie ein Benutzer mit aktivierter Selbstverwaltung seinen öffentlichen Schlüssel angeben kann.

So legen Sie Ihren öffentlichen SSH-Schlüssel fest

1. Erstellen Sie ein SSH-Schlüsselpaar.

   Die einfachste Methode besteht darin, das Schlüsselpaar lokal zu generieren. Weitere Informationen finden Sie unter How to Generate Your Own Key and Import It to Amazon EC2 (Erstellen Ihres eigenen Schlüssels und dessen Import in Amazon EC2).

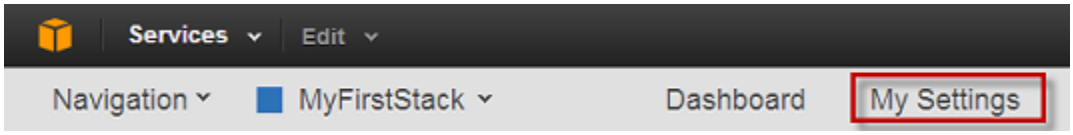
   > **Note**
   >
   > Wenn Sie PuTTYgen verwenden, um Ihr key pair zu generieren, kopieren Sie den öffentlichen Schlüssel aus dem Feld Öffentlicher Schlüssel zum Einfügen in die OpenSSH-Dateibox authorized_keys. Wenn Sie auf Öffentlichen Schlüssel speichern

klicken, wird der öffentliche Schlüssel in einem Format gespeichert, das von nicht unterstützt wird. MindTerm

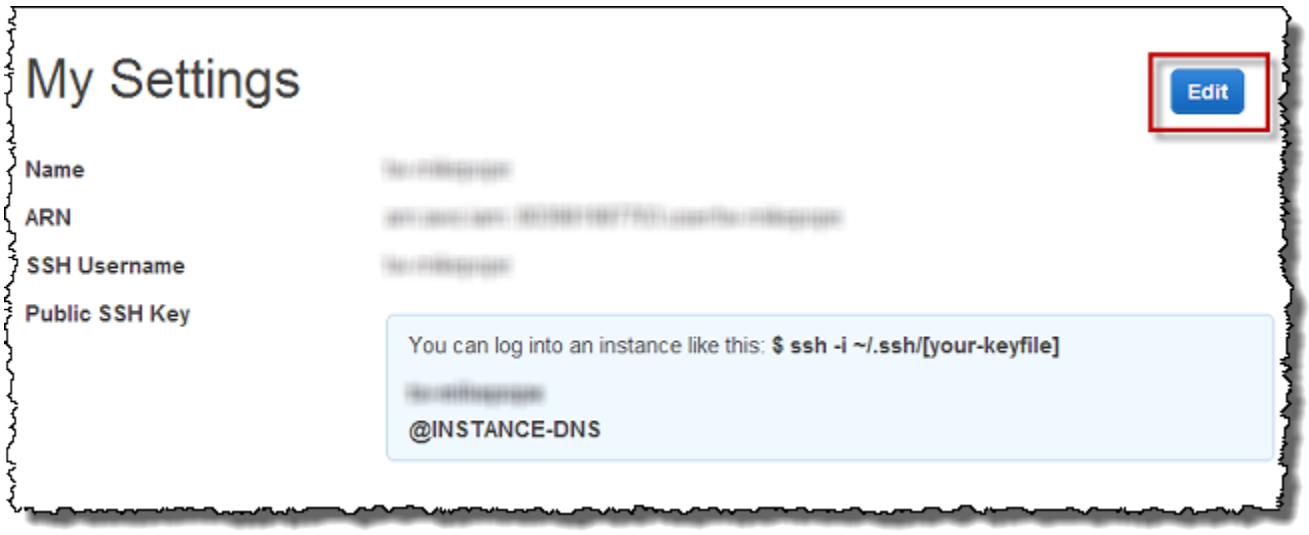
2. Melden Sie sich bei der AWS OpsWorks Stacks-Konsole als IAM-Benutzer mit aktivierter Selbstverwaltung an.

> **Important**
>
> Wenn Sie sich als Kontoinhaber oder als IAM-Benutzer anmelden, für den die Selbstverwaltung nicht aktiviert ist, zeigt AWS OpsWorks Stacks Meine Einstellungen nicht an. Wenn Sie ein Benutzer mit Administratorrechten oder der Kontoinhaber sind, können Sie stattdessen die SSH-Schlüssel festlegen, indem Sie die Seite Users (Benutzer) öffnen und die Benutzereinstellungen bearbeiten.

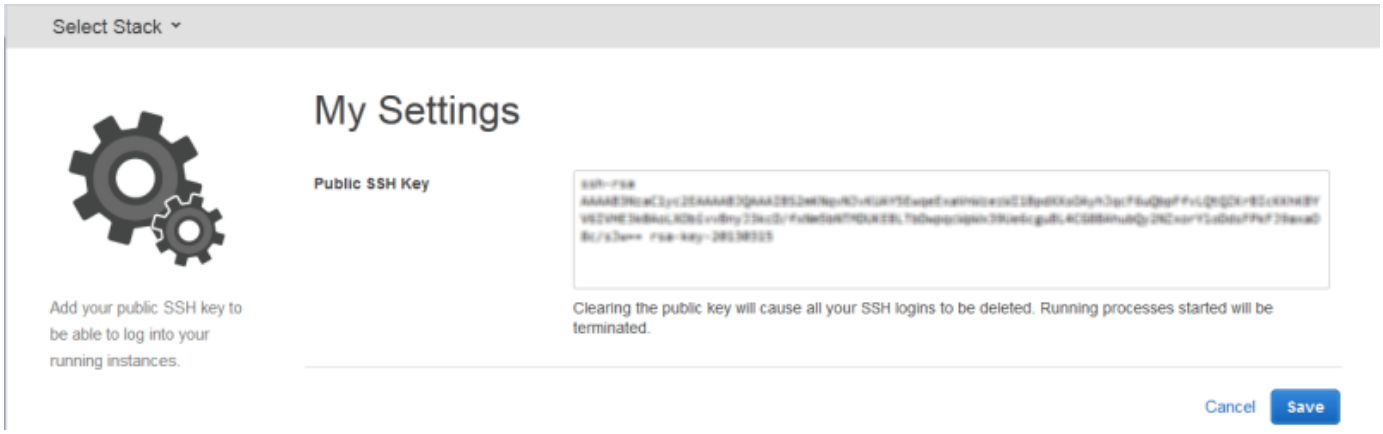
3. Wählen Sie Meine Einstellungen aus, um die Einstellungen für den angemeldeten Benutzer anzuzeigen.



4. Klicken Sie auf der Seite My Settings (Eigene Einstellungen) auf Edit (Bearbeiten).



5. Geben Sie im Feld Public SSH Key (Öffentlicher SSH-Schlüssel) Ihren öffentlichen SSH-Schlüssel ein und klicken Sie dann auf Save (Speichern).



Important

Um den integrierten MindTerm SSH-Client für die Verbindung zu Amazon EC2 EC2-Instances zu verwenden, muss ein Benutzer als IAM-Benutzer angemeldet sein und über einen öffentlichen SSH-Schlüssel verfügen, der bei Stacks registriert ist. AWS OpsWorks Weitere Informationen finden Sie unter Verwenden des integrierten SSH-Clients MindTerm .

Verwalten von Linux-Sicherheitsupdates

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sicherheits-Updates

Anbieter von Linux-Betriebssystemen liefern regelmäßige Updates. Die meisten davon sind Sicherheits-Patches für das Betriebssystem, es können aber auch Aktualisierungen für installierte Pakete sein. Sie sollten sicherstellen, dass die Betriebssysteme Ihrer Instances durch die neuesten Sicherheits-Patches auf dem aktuellen Stand sind.

Standardmäßig installiert AWS OpsWorks Stacks während der Installation automatisch die neuesten Updates, nachdem eine Instanz den Startvorgang abgeschlossen hat. AWS OpsWorks Stacks installiert Updates nicht automatisch, nachdem eine Instanz online ist, um Unterbrechungen wie den Neustart von Anwendungsservern zu vermeiden. Stattdessen verwalten Sie Updates Ihrer Online-Instances selbst, um Unterbrechungen zu minimieren.

Wir empfehlen, dass Sie einen der folgenden Schritte befolgen, um Ihre Online-Instances zu aktualisieren.

- Erstellen und starten Sie neue Instances, um Ihre aktuellen Online-Instances zu ersetzen. Löschen Sie anschließend die aktuellen Instances.

  Auf neuen Instances werden während der Einrichtung die jeweils aktuellen Sicherheits-Patches installiert.
- Führen Sie auf Linux-basierten Instances in Chef 11.10 oder älteren Stacks den Stack-Befehl Update Dependencies (Abhängigkeiten aktualisieren) aus. Hierdurch werden der aktuelle Satz von Sicherheits-Patches und andere Updates auf den angegebenen Instances installiert.

Bei beiden Ansätzen führt AWS OpsWorks Stacks das Update durch, indem es `yum update` für Amazon Linux und Red Hat Enterprise Linux (RHEL) oder `apt-get update` für Ubuntu ausgeführt wird. Jede Verteilung verarbeitet Updates etwas anders. Daher sollten Sie die Informationen in den zugehörigen Links lesen, um genau zu verstehen, wie ein Update Ihre Instances beeinflusst:

- Amazon Linux — Amazon Linux-Updates installieren Sicherheitspatches und möglicherweise auch Funktionsupdates, einschließlich Paket-Updates.

  Weitere Informationen finden Sie unter Amazon Linux AMI – Häufig gestellte Fragen.
- Ubuntu — Ubuntu-Updates beschränken sich weitgehend auf die Installation von Sicherheitspatches, können aber auch Paket-Updates für eine begrenzte Anzahl kritischer Fixes installieren.

  Weitere Informationen finden Sie unter LTS – Ubuntu Wiki.
- CentOS — CentOS-Updates behalten im Allgemeinen die Binärkompatibilität mit früheren Versionen bei.
- RHEL — RHEL-Updates behalten im Allgemeinen die Binärkompatibilität mit früheren Versionen bei.

Weitere Informationen finden Sie unter Red Hat Enterprise Linux Life Cycle (Red Hat Enterprise Linux-Lebenszyklus).

Wenn Sie mehr Kontrolle über Updates haben möchten, z. B. die Angabe bestimmter Paketversionen, können Sie automatische Updates deaktivieren, indem Sie die UpdateLayerAktionen CreateInstance, UpdateInstanceCreateLayer, oder — oder die entsprechenden AWS-SDK-Methoden oder AWS-CLI-Befehle — verwenden, um den Parameter auf zu setzen. `InstallUpdatesOnBoot` `false` Das folgende Beispiel zeigt, wie Sie mit der AWS-CLI `InstallUpdatesOnBoot` als Standardeinstellung für eine vorhandene Ebene deaktivieren können.

```
aws opsworks update-layer --layer-id layer ID --no-install-updates-on-boot
```

Sie müssen Aktualisierungen dann selbst verwalten. Sie können beispielsweise eine der folgenden Strategien nutzen:

- Implementieren Sie ein benutzerdefiniertes Rezept, das den entsprechenden Shell-Befehl ausführt, um Ihre bevorzugten Updates zu installieren.

  Da System-Updates nicht automatisch mit einem Lebenszyklusereignis verknüpft sind, schließen Sie das Rezept in Ihren benutzerdefinierten Rezeptbüchern ein, aber führen Sie sie manuell aus. Für Paket-Aktualisierungen können Sie auch die yum_package (Amazon Linux)- oder apt_package (Ubuntu)-Ressourcen anstelle eines Shell-Befehls nutzen.
- Melden Sie sich bei jeder Instance mit SSH an und führen Sie die entsprechenden Befehle manuell aus.

Verwenden von Sicherheitsgruppen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sicherheitsgruppen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Jede Amazon EC2 EC2-Instance hat eine oder mehrere zugehörige Sicherheitsgruppen, die den Netzwerkverkehr der Instance steuern, ähnlich wie bei einer Firewall. Eine Sicherheitsgruppe enthält eine oder mehrere Regeln, die jeweils eine bestimmte Kategorie des zulässigen Datenverkehrs definieren. Eine Regel definiert Folgendes:

- Den Typ des zulässigen Datenverkehrs, z. B. SSH oder HTTP.
- Das Protokoll des Datenverkehrs, z. B. TCP oder UDP.
- Den für den eingehenden Datenverkehr zulässigen IP-Adressbereich.
- Den für den Datenverkehr zulässigen Port-Bereich.

Die Sicherheitsgruppen haben zwei Arten von Regeln:

- Die Regeln für eingehenden Datenverkehr regeln den eingehenden Netzwerkverkehr.

  Zum Beispiel haben Anwendungsserver-Instances normalerweise eine Regel für den eingehenden Datenverkehr, der den eingehenden HTTP-Datenverkehr von einer beliebigen IP-Adresse an Port 80 leitet, sowie eine weitere Regel für eingehenden Datenverkehr, der SSH-Datenverkehr von einem bestimmten Satz von IP-Adressen über Port 22 zulässt.
- Die Regeln für den ausgehenden Datenverkehr steuern den ausgehenden Netzwerkverkehr.

  Üblicherweise werden die Standardeinstellung verwendet, die jeglichen ausgehenden Datenverkehr zulassen.

Weitere Informationen zu Sicherheitsgruppen finden Sie unter Amazon EC2-Sicherheitsgruppen.

Wenn Sie zum ersten Mal einen Stack in einer Region erstellen, erstellt AWS OpsWorks Stacks für jede Ebene eine integrierte Sicherheitsgruppe mit einem entsprechenden Regelsatz. Alle Gruppen verfügen über Standardregeln für den ausgehenden Datenverkehr, die sämtlichen ausgehenden Datenverkehr zulassen. Grundsätzlich lassen die Regeln für den eingehenden Datenverkehr Folgendes zu:

- Eingehender TCP-, UDP- und ICMP-Verkehr aus den entsprechenden Stacks-Schichten AWS OpsWorks
- Eingehender TCP-Datenverkehr auf Port 22 (SSH-Anmeldung).

> **Warning**
>
> Die Standardkonfiguration der Sicherheitsgruppe öffnet Port 22 (SSH) für alle Netzwerkstandorte (0.0.0.0/0). Auf diese Weise erhalten alle IP-Adressen Zugriff auf Ihre Instance über SSH. Für Produktionsumgebungen müssen Sie eine Konfiguration verwenden, die nur den SSH-Zugriff von einer bestimmten IP-Adresse oder einem bestimmte IP-Adressbereich zulässt. Aktualisieren Sie entweder die Standard-Sicherheitsgruppen unmittelbar, nachdem sie erstellt wurden, oder verwenden Sie benutzerdefinierte Sicherheitsgruppen.

- Bei Webserver-Layern muss sämtlicher TCP- und UDP-Datenverkehr über die Ports 80 (HTTP) und 443 (HTTPS) erfolgen.

> **Note**
>
> Die integrierte `AWS-OpsWorks-RDP-Server`-Sicherheitsgruppe ist allen Windows-Instances zugewiesen, um den RDP-Zugriff zuzulassen. Allerdings sind standardmäßig keine Regeln festgelegt. Wenn Sie einen Windows-Stack ausführen und mit RDP auf Instances zugreifen möchten, müssen Sie eine Regel für den eingehenden Datenverkehr hinzufügen, die den RDP-Zugriff zulässt. Weitere Informationen finden Sie unter Anmelden mit RDP.

Um die Details für jede Gruppe zu sehen, gehen Sie zur Amazon EC2 EC2-Konsole, wählen Sie im Navigationsbereich Sicherheitsgruppen und wählen Sie die Sicherheitsgruppe der entsprechenden Ebene aus. Zum Beispiel ist AWS- OpsWorks -Default-Server die standardmäßig integrierte Sicherheitsgruppe für alle Stacks, und AWS- OpsWorks - WebApp ist die standardmäßige integrierte Sicherheitsgruppe für den Chef 12-Beispielstapel.

> **Note**
>
> Wenn Sie versehentlich eine AWS OpsWorks Stacks-Sicherheitsgruppe löschen, besteht die bevorzugte Methode, sie neu zu erstellen, darin, Stacks die Aufgabe für Sie ausführen zu lassen AWS OpsWorks . Erstellen Sie einfach einen neuen Stack in derselben AWS-Region — und VPC, falls vorhanden — und AWS OpsWorks Stacks erstellt automatisch alle integrierten Sicherheitsgruppen neu, einschließlich der gelöschten. Anschließend können Sie den Stack löschen, wenn Sie keine weitere Verwendung dafür haben. Die Sicherheitsgruppen bleiben erhalten. Wenn Sie die Sicherheitsgruppe manuell neu erstellen möchten, müssen Sie eine exakte Kopie der ursprünglichen Datei unter Beachtung der Groß-/Kleinschreibung des Gruppennamens erstellen.
> Darüber hinaus versucht AWS OpsWorks Stacks, alle integrierten Sicherheitsgruppen neu zu erstellen, wenn einer der folgenden Fälle eintritt:
>
> - Sie nehmen alle Änderungen an der Einstellungsseite des Stacks in der AWS OpsWorks Stacks-Konsole vor.
> - Sie starten eine der Stack-Instances.
> - Sie erstellen einen neuen Stack.

Sie können eine der folgenden Methoden wählen, um Sicherheitsgruppen anzugeben. Sie verwenden die Einstellung OpsWorks Sicherheitsgruppen verwenden, um Ihre Präferenz anzugeben, wenn Sie einen Stack erstellen.

- Ja (Standardeinstellung) — AWS OpsWorks Stacks ordnet jeder Ebene automatisch die entsprechende integrierte Sicherheitsgruppe zu.

  Sie können eine Feinabstimmung der integrierten Sicherheitsgruppe eines Layers vornehmen, indem Sie eine benutzerdefinierte Sicherheitsgruppe mit Ihren bevorzugten Einstellungen hinzufügen. Wenn Amazon EC2 jedoch mehrere Sicherheitsgruppen auswertet, verwendet es die am wenigsten restriktiven Regeln, sodass Sie diesen Ansatz nicht verwenden können, um restriktivere Regeln als die integrierte Gruppe anzugeben.
- Nein — AWS OpsWorks Stacks ordnet integrierte Sicherheitsgruppen keinen Ebenen zu.

  Sie müssen geeignete Sicherheitsgruppen erstellen und alle von Ihnen erstellten Layer mindestens einer Sicherheitsgruppe zuordnen. Verwenden Sie diese Methode, um restriktivere Regeln als die integrierten Gruppen festzulegen. Beachten Sie dabei, dass Sie eine integrierte Sicherheitsgruppe

immer noch manuell einem Layer zuordnen können, wenn Sie das bevorzugen. Benutzerdefinierte Sicherheitsgruppen sind nur für die Layer erforderlich, für die benutzerdefinierte Einstellungen anzugeben sind.

> **Important**
>
> Wenn Sie integrierte Sicherheitsgruppen verwenden, ist es nicht möglich, durch manuelles Ändern der Gruppe restriktivere Regeln zu erstellen. Jedes Mal, wenn Sie einen Stack erstellen, überschreibt AWS OpsWorks Stacks die Konfigurationen der integrierten Sicherheitsgruppen, sodass alle Änderungen, die Sie vornehmen, verloren gehen, wenn Sie das nächste Mal einen Stack erstellen. Wenn für eine Ebene restriktivere Sicherheitsgruppeneinstellungen erforderlich sind als für die integrierte Sicherheitsgruppe, setzen Sie OpsWorks Sicherheitsgruppen verwenden auf Nein, erstellen Sie benutzerdefinierte Sicherheitsgruppen mit Ihren bevorzugten Einstellungen und weisen Sie sie den Ebenen bei der Erstellung zu.

AWS OpsWorks Stacks-Unterstützung für Chef 12 Linux

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Dieser Abschnitt bietet einen kurzen Überblick über AWS OpsWorks Stacks für Chef 12 Linux. Weitere Informationen zu Chef 12 für Windows finden Sie unter Erste Schritte: Windows. Informationen zu vorherigen Chef-Versionen für Linux finden Sie unter Chef 11.10 und früheren Versionen für Linux.

Übersicht

AWS OpsWorks Stacks unterstützt Chef 12, die neueste Version von Chef, für Linux-Stacks. Weitere Informationen finden Sie unter Learn Chef.

AWS OpsWorks Stacks unterstützt weiterhin Chef 11.10 für Linux-Stacks. Wenn Sie jedoch fortgeschrittener Benutzer von Chef sind und die Vorteile der großen Auswahl an Community-Rezeptbüchern nutzen oder eigene benutzerdefinierte Rezeptbücher schreiben möchten, empfehlen wir die Verwendung von Chef 12. Chef 12-Stacks bieten die folgenden Vorteile gegenüber Chef 11.10 und früheren Stacks für Linux:

- Zwei separate Chef-Läufe — Wenn ein Befehl auf einer Instance ausgeführt wird, führt der AWS OpsWorks Stacks-Agent jetzt zwei isolierte Chef-Läufe aus: einen Lauf für Aufgaben, die die Instance in andere AWS-Services wie AWS Identity and Access Management (IAM) integrieren, und einen Lauf für Ihre benutzerdefinierten Kochbücher. Beim ersten Chef-Lauf wird der AWS OpsWorks Stacks-Agent auf der Instance installiert und Systemaufgaben wie Benutzereinrichtung und -verwaltung, Einrichtung und Konfiguration von Volumes, Konfiguration von CloudWatch Metriken usw. ausgeführt. In der zweiten Ausführung werden ausschließlich Ihre benutzerdefinierten Rezepte für AWS OpsWorks Stapelt Lifecycle-Ereignisse ausgeführt. Diese zweite Ausführung ermöglicht Ihnen, Ihre eigenen Chef-Rezeptbücher oder Community-Rezeptbücher zu verwenden.
- Auflösung von Namespace-Konflikten: Vor Chef 12 führte AWS OpsWorks Stacks Systemaufgaben durch und führte integrierte und benutzerdefinierte Rezepte in einer gemeinsamen Umgebung aus. Dies führte zu Namespace-Konflikten und mangelnder Klarheit darüber, welche Rezepte AWS OpsWorks Stacks ausgeführt hatte. Unerwünschte Standardkonfigurationen mussten manuell überschrieben werden – eine zeitaufwendige und fehleranfällige Aufgabe. In Chef 12 für Linux unterstützt AWS OpsWorks Stacks keine integrierten Chef-Kochbücher mehr für Anwendungsserverumgebungen wie PHP, Node.js oder Rails. Durch den Wegfall integrierter Rezepte beseitigt AWS OpsWorks Stacks das Problem der Namenskollisionen zwischen integrierten und benutzerdefinierten Rezepten.
- Starke Unterstützung für Kochbücher der Chef-Community — AWS OpsWorks Stacks Chef 12 Linux bietet mehr Kompatibilität und Unterstützung für Community-Kochbücher aus dem Chef-Supermarkt. Sie können jetzt Community-Kochbücher verwenden, die den integrierten Kochbüchern, die AWS OpsWorks Stacks zuvor bereitgestellt hat, überlegen sind — Kochbücher, die für die Verwendung mit den neuesten Anwendungsserverumgebungen und Frameworks konzipiert sind. Sie können die meisten dieser Rezeptbücher ohne Änderungen auf Chef 12 für Linux ausführen. Weitere Informationen finden Sie unter Chef Supermarket auf der Learn Chef-Website, der Chef Supermarket-Website und im Chef Cookbooks-Repository unter. GitHub
- Rechtzeitige Chef 12-Updates — AWS OpsWorks Stacks wird seine Chef-Umgebung kurz nach jeder Chef-Veröffentlichung auf die neueste Chef 12-Version aktualisieren. Mit Chef 12 werden kleinere Chef-Updates und neue AWS OpsWorks Stacks-Agentenversionen zusammenfallen.

Dadurch können Sie neue Chef-Versionen direkt testen. Außerdem können Sie für Ihre Chef-Rezepte und -Anwendungen die Vorteile der neuesten Chef-Funktionen nutzen.

Weitere Informationen über unterstützte Chef-Versionen vor Chef 12 finden Sie unter Chef 11.10 und früheren Versionen für Linux.

Wechsel zu Chef 12

Die wichtigsten AWS OpsWorks Stacks-Änderungen für Chef 12 Linux im Vergleich zur Unterstützung früherer Chef-Versionen 11.10, 11.4 und 0.9 lauten wie folgt:

- Integrierte Layer werden für Chef 12 für Linux-Stacks nicht mehr bereitgestellt oder unterstützt. Da nur Ihre benutzerdefinierten Rezepte ausgeführt werden, besteht durch das Wegfallen dieser Unterstützung nun totale Transparenz dahingehend, wie die Instance eingerichtet ist. Zudem wird das Schreiben und Verwalten benutzerdefinierter Rezeptbücher vereinfacht. Beispielsweise ist es nicht mehr erforderlich, Attribute der integrierten Stacks-Rezepte zu überschreiben. AWS OpsWorks Durch das Entfernen der integrierten Ebenen kann AWS OpsWorks Stacks auch Kochbücher, die von der Chef-Community entwickelt und verwaltet werden, besser unterstützen, sodass Sie sie in vollem Umfang nutzen können. Die integrierten Ebenentypen, die in Chef 12 für Linux nicht mehr verfügbar sind, sind: AWS Flow (Ruby), Ganglia, HAProxy, Java App Server, Memcached, MySQL, Node.js App Server, PHP App Server, Rails App Server und Static Web Server.
  - Da AWS OpsWorks Stacks die von Ihnen bereitgestellten Rezepte ausführt, ist es nicht mehr erforderlich, die integrierten AWS OpsWorks Stacks-Attribute durch das Ausführen benutzerdefinierter Kochbücher zu überschreiben. Um Attribute in Ihren eigenen Rezepten oder Community-Rezepten zu überschreiben, folgen Sie den Anweisungen und Beispielen unter About Attributes in der Chef 12-Dokumentation.
- AWS OpsWorks Stacks unterstützt weiterhin die folgenden Ebenen für Chef 12 Linux-Stacks:
  - Benutzerspezifische Layers
  - Amazon RDS-Serviceschicht
  - ECS-Cluster-Ebenen
- Die Stack-Konfiguration und Data Bags für Chef 12 Linux wurden geändert und sehen ihren Entsprechungen für Chef 12.2 für Windows sehr ähnlich. Dadurch können Sie diese Data Bags leichter abfragen, analysieren und Probleme beheben, vor allem wenn Sie mit Stacks mit verschiedenen Arten von Betriebssystemen arbeiten. Beachten Sie, dass AWS OpsWorks Stacks keine verschlüsselten Datenbeutel unterstützt. Um vertrauliche Daten in verschlüsselter Form zu

speichern, wie z. B. Passwörter oder Zertifikate, empfehlen wir, diese in einem privaten S3-Bucket zu speichern. Anschließend können Sie ein benutzerdefiniertes Rezept erstellen, das zum Abrufen der Daten das Amazon SDK für Ruby verwendet. Ein Beispiel finden Sie unter Verwenden des -SDK for Ruby. Weitere Informationen finden Sie unter AWS OpsWorks Referenz für Stacks Data Bag.

- In Chef 12 Linux ist Berkshelf nicht mehr auf Stack-Instances installiert. Stattdessen empfehlen wir die Verwendung von Berkshelf auf einem lokalen Entwicklungscomputer, um Ihre Rezeptbuch-Abhängigkeiten lokal zu verpacken. Laden Sie dann Ihr Paket einschließlich der Abhängigkeiten auf Amazon Simple Storage Service hoch. Als letzten Schritt ändern Sie Ihren Chef 12 Linux-Stack so ab, dass das hochgeladene Paket als Rezeptbuchquelle verwendet wird. Weitere Informationen finden Sie unter Lokales Verpacken von Rezeptbuch-Abhängigkeiten.
- RAID-Konfigurationen für EBS-Volumes werden nicht mehr unterstützt. Für eine höhere Leistung können Sie bereitgestellte IOPS für Amazon Elastic Block Store (Amazon EBS) verwenden.
- Autofs wird nicht mehr unterstützt.
- Subversion-Repositorys werden nicht mehr unterstützt.
- Betriebssystem-Paketinstallationen pro Layer müssen jetzt mit benutzerdefinierten Rezepten durchgeführt werden. Weitere Informationen finden Sie unter Paketinstallationen pro Layer.

Unterstützte Betriebssysteme

Chef 12 unterstützt dieselben Linux-Betriebssysteme wie vorherige Versionen von Chef. Eine Liste der Typen und Versionen von Linux-Betriebssystemen, die Chef 12 Linux-Stacks verwenden können, finden Sie unter Linux-Betriebssysteme.

Unterstützte Instance-Typen

AWS OpsWorks Stacks unterstützt alle Instance-Typen für Chef 12-Linux-Stacks mit Ausnahme spezialisierter Instance-Typen wie High Performance Computing (HPC) -Cluster-Computing, Cluster-GPU und High-Memory-Cluster-Instance-Typen.

Weitere Informationen

Weitere Informationen zum Arbeiten mit Chef 12 für Linux-Stacks finden Sie in den folgenden Themen:

- Erste Schritte: Beispiel

Stellt Ihnen AWS OpsWorks Stacks vor und führt Sie durch eine kurze praktische Übung mit der AWS OpsWorks Stacks-Konsole zur Erstellung einer Node.js -Anwendungsumgebung.

- Erste Schritte: Linux

  Führt Sie in AWS OpsWorks Stacks und Chef 12 Linux ein und führt Sie durch eine praktische Übung mit der AWS OpsWorks Stacks-Konsole, um einen grundlegenden Chef 12-Linux-Stack zu erstellen, der eine einfache Ebene mit einer Node.js -App enthält, die den Datenverkehr bedient.

- Benutzerspezifische Layers

  Enthält Anleitungen für das Hinzufügen eines Layers, der Rezeptbücher und Rezepte enthält, zu einem Chef 12 Linux-Stack. Sie können sofort verfügbare Rezeptbücher und Rezepte verwenden, die die Chef-Community bereitstellt, oder Sie können Ihre eigenen erstellen.

- Wechsel zu Data Bags

  Vergleicht Instance-JSON-Daten, die von Linux-Stacks verwendet werden, auf denen Chef 11 und frühere Versionen ausgeführt werden, mit Instance-JSON-Daten, die von Linux-Stacks verwendet werden, auf denen Chef 12 ausgeführt wird, und stellt diese einander gegenüber. Hier finden Sie auch Verweise auf Referenzdokumentation zum Instance-JSON-Format von Chef 12.

Wechsel der Stack-Einstellungen von Attributen zu Data Bags

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks bietet eine Vielzahl von Stapeleinstellungen für Ihre Chef-Rezepte. Diese Stack-Einstellungen umfassen Werte, wie:

- Quell-URLs der Stack-Rezeptbücher
- Layer-Volume-Konfigurationen
- Instance-Hostnamen

- Elastic Load Balancing DNS-Namen
- Quell-URLs der Anwendungen
- Benutzernamen

Aus Rezepten auf Stack-Einstellungen zu verweisen, sorgt für einen stabileren Rezeptcode und eine geringere Fehleranfälligkeit als festkodierte Stack-Einstellungen direkt in den Rezepten. In diesem Abschnitt wird beschrieben, wie Sie auf diese Stack-Einstellungen zugreifen und wie Sie den Wechsel von Attributen in Chef 11.10 und früheren Versionen für Linux zu Data Bags in Chef 12 Linux vollziehen.

In Chef 11.10 und früheren Versionen für Linux stehen Stack-Einstellungen als Chef-Attribute zur Verfügung und der Zugriff erfolgt über das `node`-Chef-Objekt oder über die Chef-Suche. Diese Attribute werden auf AWS OpsWorks Stacks-Instanzen in einer Reihe von JSON-Dateien im `/var/lib/aws/opsworks/chef` Verzeichnis gespeichert. Weitere Informationen finden Sie unter Stack-Konfigurations- und Bereitstellungsattribute: Linux.

In Chef 12 Linux sind Stack-Einstellungen als Chef-Data Bags verfügbar und der Zugriff erfolgt nur über die Chef-Suche. Datentaschen werden auf AWS OpsWorks Stacks-Instanzen in einer Reihe von JSON-Dateien im `/var/chef/runs/`*`run-ID`*`/data_bags` Verzeichnis gespeichert, wobei *`Run-ID eine eindeutige ID`* ist, die AWS OpsWorks Stacks jedem Chef-Lauf auf einer Instanz zuweist. Stack-Einstellungen sind nicht mehr als Chef-Attribute verfügbar, sodass auf Stack-Einstellungen nicht mehr über das Chef-Objekt `node` zugegriffen werden kann. Weitere Informationen hierzu finden Sie unter AWS OpsWorks Referenz für Stacks Data Bag.

In Chef 11.10 und früheren Versionen für Linux verwendet der folgende Rezept-Code zum Beispiel das Chef-Objekt `node`, um Attribute abzurufen, die den Kurznamen und die Quell-URL einer Anwendung repräsentieren. Dann schreibt er mithilfe des Chef-Protokolls diese zwei Attributwerte:

```
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 verwendet der folgende Rezept-Code den Suchindex `aws_opsworks_app`, um die Inhalte des ersten Data Bag-Elements im Data Bag `aws_opsworks_app` abzurufen. Der Code schreibt dann zwei Nachrichten in das Chef-Protokoll, eine mit dem Kurznamen der Data Bag-Inhalte der Anwendung und eine andere mit der Quell-URL der Data Bag-Inhalte der Anwendung:

```
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']}' **********")
```

Wenn Sie den Rezept-Code, der auf Stack-Einstellungen von Chef 11.10 und früheren Versionen für Linux zugreift, auf Chef 12 für Linux migrieren wollen, müssen Sie den Code wie folgt ändern:

- Greifen Sie auf Chef-Data Bags statt auf Chef-Attribute zu.
- Verwenden Sie die Chef-Suche anstelle des Chef-Objekts `node`.
- Verwenden Sie AWS OpsWorks Stacks-Datentaschennamen wie`aws_opsworks_app`, anstatt AWS OpsWorks Stacks-Attributnamen wie und zu verwenden. `opsworks` `deploy`

Weitere Informationen hierzu finden Sie unter AWS OpsWorks Referenz für Stacks Data Bag.

Support für frühere Chef-Versionen in AWS OpsWorks Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Dieser Abschnitt bietet einen kurzen Überblick über die AWS OpsWorks Stacks-Dokumentation für frühere Chef-Versionen.

Chef 11.10 und früheren Versionen für Linux

Enthält Dokumentation zur AWS OpsWorks Stacks-Unterstützung für Chef 11.10, 11.4 und 0.9 für Linux-Stacks.

Chef 11.10 und früheren Versionen für Linux

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Dieser Abschnitt bietet einen kurzen Überblick über die AWS OpsWorks Stacks-Dokumentation für Chef 11.10, 11.4 und 0.9 für Linux.

Erste Schritte mit Chef 11 Linux-Stacks

Enthält eine Anleitung zur Erstellung eines einfachen, aber funktionellen PHP-Anwendungsserver-Stacks.

Erstellen Ihres ersten Node.js-Stacks

Beschreibt, wie Sie einen Linux-Stack erstellen, der einen Node.js-Anwendungsserver unterstützt, und wie eine einfache Anwendung bereitgestellt wird.

Stacks anpassen AWS OpsWorks

Beschreibt, wie Sie AWS OpsWorks Stacks an Ihre spezifischen Anforderungen anpassen können.

Rezeptbücher 101

Beschreibt, wie Rezepte für AWS OpsWorks Stacks-Instanzen implementiert werden.

Load Balancing eines Layers

Beschreibt, wie die verfügbaren AWS OpsWorks Stacks-Load-Balancing-Optionen verwendet werden.

Ausführen eines Stacks in einer VPC

Beschreibt das Erstellen und Ausführen eines Stacks in einer Virtual Privat Cloud.

Migration von Chef-Server

Enthält Richtlinien für die Migration von Chef Server zu AWS OpsWorks Stacks.

[AWS OpsWorks Stacks-Ebenenreferenz]

Beschreibt die verfügbaren integrierten AWS OpsWorks Stacks-Ebenen.

[Bestandteile eines Rezeptbuchs]

Beschreibt die drei Standardkomponenten des Rezeptbuchs: Attribute, Vorlagen und Rezepte.

[Stack-Konfigurations- und Bereitstellungsattribute: Linux]

Beschreibt die Stack-Konfigurations- und Bereitstellungsattribute für Linux.

[Integrierte Rezeptbuchattribute]

Beschreibt, wie integrierte Rezeptattribute zum Steuern der Konfiguration der installierten Software verwendet werden.

[Beheben von Chef 11.10 und früheren Versionen für Linux]

Beschreibt Ansätze zur Behebung verschiedener Probleme in AWS OpsWorks Stacks.

Erste Schritte mit Chef 11 Linux-Stacks

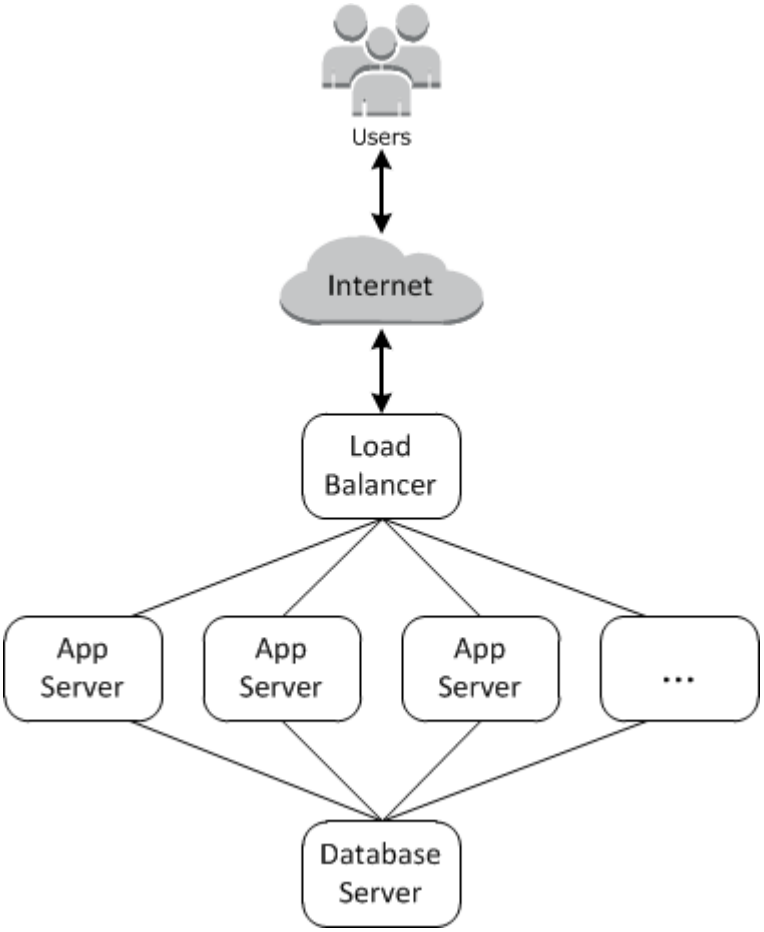
Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

In diesem Abschnitt werden die ersten Schritte zur Verwendung von Linux-Stacks mit Chef 11 beschrieben. Weitere Informationen zu den ersten Schritten zur Verwendung von Chef 12 Linux-Stacks finden Sie unter Erste Schritte: Linux. Weitere Informationen zu den ersten Schritten zur Verwendung von Chef 12 Windows-Stacks finden Sie unter Erste Schritte: Windows.

Cloud-basierte Anwendungen erfordern in der Regel eine Gruppe verwandter Ressourcen — Anwendungsserver, Datenbankserver usw. —, die gemeinsam erstellt und verwaltet werden müssen. Diese Instances-Sammlung wird Stack genannt. Ein einfacher Anwendungs-Stack hat beispielsweise folgende Struktur.



Die grundlegende Architektur umfasst Folgendes:

- Einen Load Balancer zur gleichmäßigen Verteilung des eingehenden Datenverkehrs von Benutzern auf die Anwendungsserver.
- So viele Anwendungsserver-Instances wie erforderlich, um den Datenverkehr handhaben zu können.
- Einen Datenbankserver als Backend-Datenspeicher für die Anwendungsserver.

Darüber hinaus benötigen Sie in der Regel eine Möglichkeit zur Verteilung von Anwendungen auf die Anwendungsserver, Überwachung des Stacks usw.

AWS OpsWorks Stacks bietet eine einfache und unkomplizierte Möglichkeit, Stacks und die zugehörigen Anwendungen und Ressourcen zu erstellen und zu verwalten. In diesem Kapitel werden die Grundlagen von AWS OpsWorks Stacks — zusammen mit einigen der komplexeren Funktionen — vorgestellt, indem Sie im Diagramm Schritt für Schritt durch den Prozess der Erstellung des Anwendungsserver-Stacks geführt werden. Es verwendet ein inkrementelles Entwicklungsmodell, das mit AWS OpsWorks Stacks leicht nachzuvollziehen ist: Richten Sie einen Basis-Stack ein und fügen Sie, sobald er ordnungsgemäß funktioniert, Komponenten hinzu, bis Sie eine Implementierung mit vollem Funktionsumfang erhalten.

- Schritt 1: Erfüllen der Voraussetzungen erläutert die vorbereitenden Maßnahmen, um mit der Anleitung zu beginnen.
- Schritt 2: Erstellen eines einfachen Anwendungsserver-Stacks – Chef 11 erläutert die Einrichtung eines minimalen Stacks bestehend aus einem einzigen Anwendungsserver.
- Schritt 3: Hinzufügen eines Backend-Datenspeichers erläutert, wie Sie einen Datenbankserver hinzufügen und diesen mit dem Anwendungsserver verbinden können.
- Schritt 4: Skalieren MyStack erläutert, wie Sie einen Stack skalieren können, um durch Hinzufügen von weiteren Anwendungsservern und einem Load Balancer zur Verteilung des eingehenden Datenverkehrs höhere Lasten verarbeiten zu können.

Themen

Schritt 1: Erfüllen der Voraussetzungen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Um mit der Anleitung beginnen zu können, müssen Sie die folgenden Einrichtungsschritte ausführen. Zu diesen Einrichtungsschritten gehören die Registrierung für ein AWS Konto, die Erstellung eines Administratorbenutzers und die Zuweisung von Zugriffsberechtigungen für Stacks. AWS OpsWorks

Wenn Sie bereits eine der Erste Schritte mit AWS OpsWorks Stacks-Anleitungen durchgeführt haben, erfüllen Sie die Voraussetzungen für diese Anleitung. Sie können diesen Schritt überspringen und Schritt 2: Erstellen eines einfachen Anwendungsserver-Stacks – Chef 11 aufrufen.

Themen

- Registriere dich für ein AWS-Konto
- Erstellen Sie einen Benutzer mit Administratorzugriff
- Weisen Sie Ihrem Benutzer Dienstzugriffsberechtigungen zu

Registriere dich für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu https://aws.amazon.com/ auf und klicken Sie auf Mein Konto.

Erstellen Sie einen Benutzer mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen Sie einen Benutzer mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis im AWS IAM Identity Center Benutzerhandbuch.

Melden Sie sich als Benutzer mit Administratorzugriff an

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal.

Weisen Sie weiteren Benutzern Zugriff zu

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

   Anweisungen finden Sie im Benutzerhandbuch unter Einen Berechtigungssatz erstellen.AWS IAM Identity Center
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Anweisungen finden Sie im AWS IAM Identity Center Benutzerhandbuch unter Gruppen hinzufügen.

Weisen Sie Ihrem Benutzer Dienstzugriffsberechtigungen zu

Ermöglichen Sie den Zugriff auf den AWS OpsWorks Stacks-Dienst (und die zugehörigen Dienste, auf die AWS OpsWorks Stacks angewiesen ist), indem Sie Ihrer Rolle oder Ihrem `AmazonS3FullAccess` Benutzer die Berechtigungen `AWSOpsWorks_FullAccess` und hinzufügen.

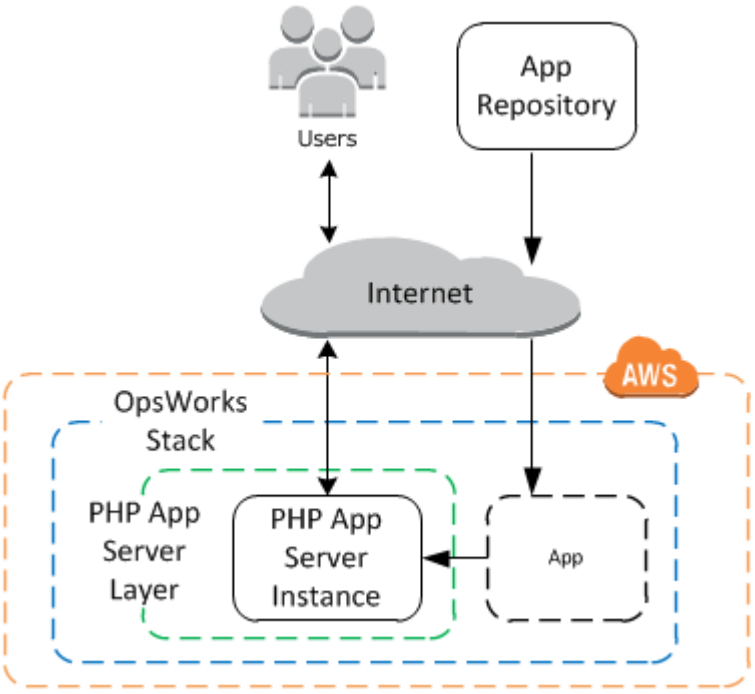
Weitere Informationen zum Hinzufügen von Berechtigungen finden Sie unter Hinzufügen von IAM-Identitätsberechtigungen (Konsole).

Nun haben Sie alle Einrichtungsschritte abgeschlossen und können mit dieser Anleitung beginnen.

Schritt 2: Erstellen eines einfachen Anwendungsserver-Stacks – Chef 11

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Ein Basis-Anwendungsserver-Stack besteht aus einer einzelnen Anwendungsserver-Instance mit einer öffentlichen IP-Adresse für den Empfang von Benutzeranforderungen. Der Anwendungscode und zugehörige Dateien werden in einem separaten Repository gespeichert und von dort auf dem Server bereitgestellt. Das folgende Diagramm veranschaulicht einen solchen Stack.



Der Stack besteht aus folgenden Komponenten:

- Einer Ebene, die eine Gruppe von Instances repräsentiert und festlegt, wie diese konfiguriert werden.

  Die Ebene in diesem Beispiel stellt eine Gruppe von PHP App Server-Instanzen dar.
- Eine Instance, die eine Amazon EC2 EC2-Instance darstellt.

  In diesem Fall wird die Instance konfiguriert, um einen PHP-Anwendungsserver zu betreiben. Ebenen können eine beliebige Anzahl von Instanzen haben. AWS OpsWorks Stacks unterstützt auch mehrere andere App-Server. Weitere Informationen finden Sie unter Anwendungsserverebene.
- Eine Anwendung, die die erforderlichen Informationen enthält, um eine Anwendung auf dem Anwendungsserver zu installieren.

  Der Code wird in einem Remote-Repository gespeichert, z. B. in einem Git-Repository oder einem Amazon S3 S3-Bucket.

In den folgenden Abschnitten wird beschrieben, wie Sie die AWS OpsWorks Stacks-Konsole verwenden, um den Stack zu erstellen und die Anwendung bereitzustellen. Sie können auch eine AWS CloudFormation Vorlage verwenden, um einen Stack bereitzustellen. Eine Beispielvorlage, die den in diesem Thema beschriebenen Stack bereitstellt, finden Sie unter OpsWorks AWS-Snippets.

Themen

Schritt 2.1: Erstellen eines Stacks – Chef 11

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie starten ein AWS OpsWorks Stacks-Projekt, indem Sie einen Stack erstellen, der als Container für Ihre Instances und andere Ressourcen fungiert. Die Stack-Konfiguration legt einige grundlegende Einstellungen fest, wie z. B. die AWS-Region und das Standardbetriebssystem, die von allen Stack-Instances gemeinsam genutzt werden.

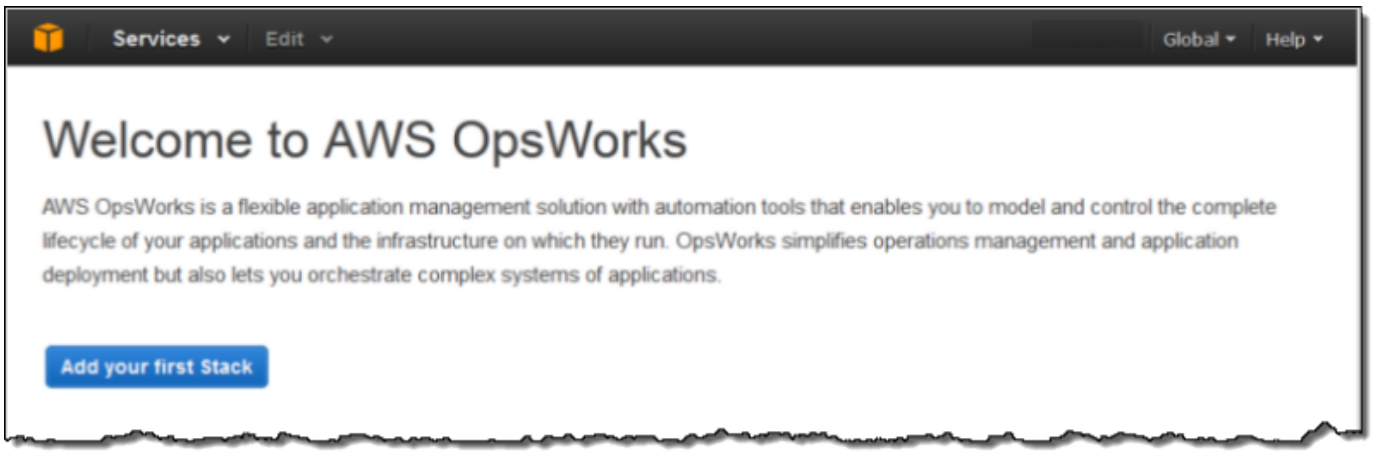
> **Note**
>
> Diese Seite unterstützt Sie beim Erstellen Chef 11-Stacks. Weitere Informationen zum Erstellen von Chef 12-Stacks finden Sie unter Erstellen eines Stacks.

Diese Seite unterstützt Sie beim Erstellen von Chef 11-Stacks.

Erstellen eines neuen Stacks

1. Hinzufügen eines Stacks

   Melden Sie sich bei der AWS OpsWorks Stacks-Konsole an. Wenn für das Konto keine vorhandenen Stacks vorhanden sind, wird die OpsWorks Seite Willkommen bei AWS angezeigt. Klicken Sie auf Add your first stack. Andernfalls wird das AWS OpsWorks Stacks-Dashboard angezeigt, in dem die Stacks Ihres Kontos aufgeführt sind. Klicken Sie auf Stack hinzufügen.



2. Konfigurieren des Stacks

   Wählen Sie auf der Seite Add Stack (Stack hinzufügen) die Option Chef 11 stack (Chef 11-Stack) aus und geben Sie die folgenden Einstellungen an:

   Stack name

   Geben Sie einen Namen für Ihren Stack ein, der alphanumerische Zeichen (a—z, A—Z und 0—9) und Bindestriche (-) enthalten kann. Der Beispiel-Stack in dieser Anleitung hat den Namen **`MyStack`**.

   Region

   Wählen Sie US West (Oregon) als Region des Stacks aus.

   Übernehmen Sie die Standardwerte für die restlichen Einstellungen und klicken Sie auf Add Stack (Stack hinzufügen). Weitere Informationen zu den verschiedenen Stack-Einstellungen finden Sie unter Erstellen eines neuen Stacks.

Schritt 2.2: Fügen Sie einen PHP-App-Serverlayer hinzu - Chef 11

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Obwohl ein Stack im Grunde ein Container für Instances ist, fügen Sie einem Stack Instances nicht direkt hinzu. Fügen Sie einen Layer hinzu, der eine Gruppe verwandter Instances repräsentiert, und fügen Sie dem Layer dann Instances hinzu.

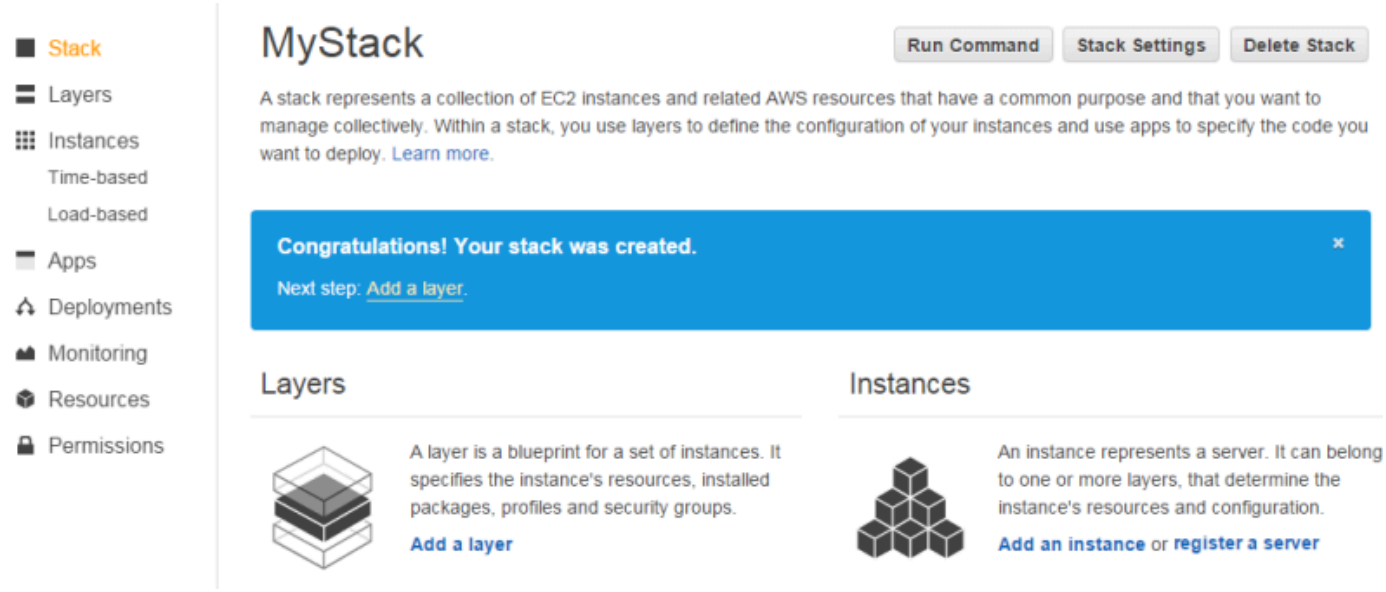
Eine Ebene ist im Grunde eine Blaupause, die AWS OpsWorks Stacks verwendet, um eine Reihe von Amazon EC2 EC2-Instances mit derselben Konfiguration zu erstellen. Sie fügen dem Stack einen Layer für jede Gruppe verwandter Instances hinzu. AWS OpsWorks Stacks enthält eine Reihe integrierter Ebenen, um Gruppen von Instanzen darzustellen, auf denen Standard-Softwarepakete wie ein MySQL-Datenbankserver oder ein PHP-Anwendungsserver ausgeführt werden. Darüber hinaus können Sie teilweise oder vollständig angepasste Layers erstellen, die Ihren spezifischen Anforderungen entsprechen. Weitere Informationen finden Sie unter Stacks anpassen AWS OpsWorks.

MyStack hat eine Ebene, die integrierte PHP App Server-Schicht, die eine Gruppe von Instanzen darstellt, die als PHP-Anwendungsserver fungieren. Weitere Informationen, einschließlich der Beschreibungen der integrierten Layer finden Sie unter Ebenen.

Um einen PHP-App-Server-Layer hinzuzufügen MyStack

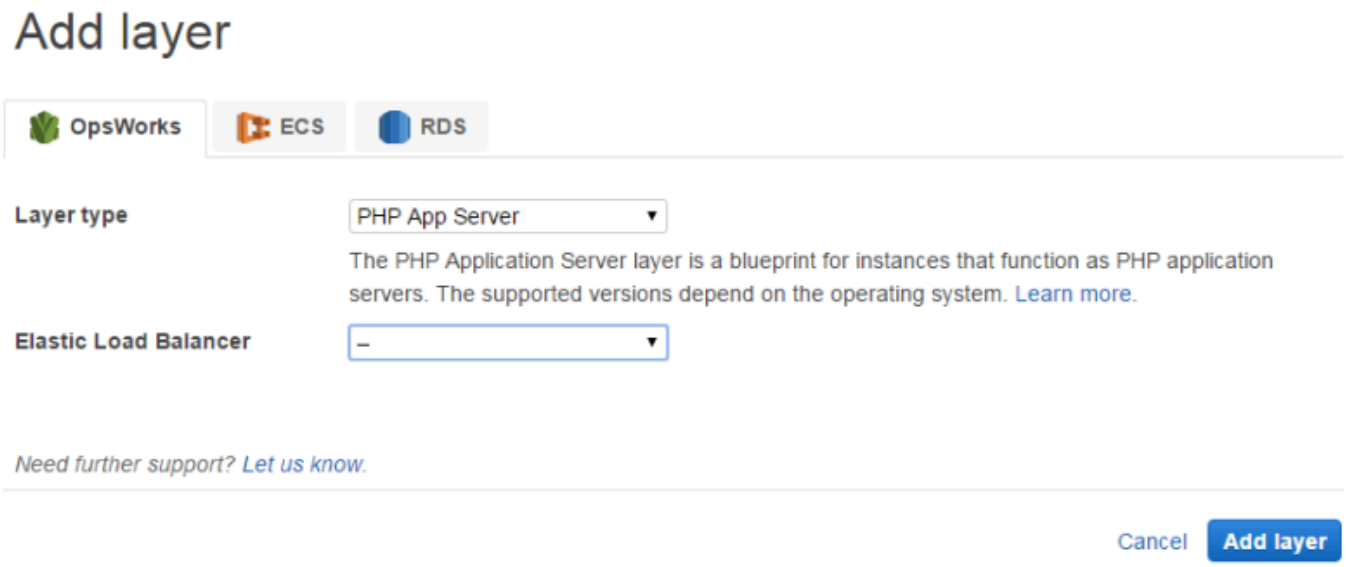
1. Öffnen der Seite "Layer hinzufügen"

   Nachdem Sie den Stack erstellt haben, zeigt AWS OpsWorks Stacks die Stack-Seite an. Klicken Sie auf Add a layer (Ebene hinzufügen), um Ihre erste Ebene hinzufügen.



2. Spezifizieren eines Layer-Typs und Konfigurieren des Layers

   Wählen Sie im Feld Layer-Typ die Option PHP App Server aus, akzeptieren Sie die Elastic Load Balancer Balancer-Standardeinstellung und klicken Sie auf Layer hinzufügen. Nachdem Sie die Ebene erstellt haben, können Sie andere Attribute festlegen, wie z. B. die EBS-Volume-Konfiguration durch Bearbeiten der Ebene.



Schritt 2.3: Fügen Sie dem PHP App Server Layer eine Instanz hinzu — Chef 11

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Eine AWS OpsWorks Stacks-Instance steht für eine bestimmte Amazon EC2 EC2-Instance:

- Die Konfiguration der Instance spezifiziert einige Grundlagen wie das Amazon EC2-Betriebssystem und die Größe; sie läuft, macht aber nicht viel.
- Der Layer der Instance fügt dieser Funktionen hinzu, indem er z.B. festlegt, welche Pakete installiert werden und ob die Instance eine elastische IP-Adresse hat.

AWS OpsWorks Stacks installiert auf jeder Instance, die mit dem Service interagiert, einen Agenten. Um einer Instanz die Funktionalität einer Ebene hinzuzufügen, weist AWS OpsWorks Stacks den Agenten an, kleine Anwendungen, sogenannte Chef-Rezepte, auszuführen, mit denen Anwendungen und Pakete installiert, Konfigurationsdateien erstellt usw. werden können. AWS OpsWorks Stacks führt Rezepte an wichtigen Punkten im Lebenszyklus der Instanz aus. OpsWorks Führt beispielsweise Setup-Rezepte aus, nachdem die Instanz den Startvorgang abgeschlossen hat, um Aufgaben wie die Installation von Software zu erledigen, und führt Deploy-Rezepte aus, wenn Sie eine App bereitstellen, um den Code und die zugehörigen Dateien zu installieren.

> **Note**
>
> Falls Sie wissen möchten, wie die Rezepte funktionieren, finden Sie alle in AWS OpsWorks Stacks integrierten Rezepte in einem öffentlichen GitHub Repository: OpsWorks Cookbooks. Sie können auch Ihre eigenen benutzerdefinierten Rezepte erstellen und diese von AWS OpsWorks Stacks ausführen lassen, wie weiter unten beschrieben.

Um einen PHP-Anwendungsserver hinzuzufügen MyStack, fügen Sie dem PHP App Server-Layer, den Sie im vorherigen Schritt erstellt haben, eine Instanz hinzu.

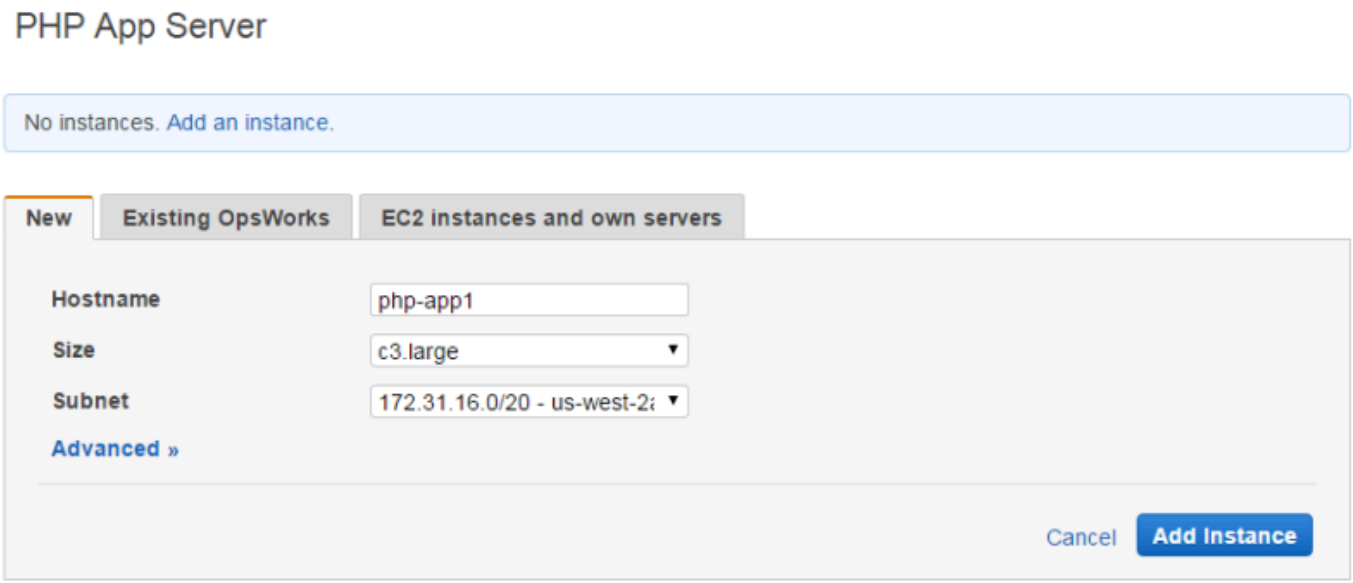
Um dem PHP App Server-Layer eine Instanz hinzuzufügen

1. Öffnen von "Instance hinzufügen"

   Nachdem Sie die Ebene hinzugefügt haben, zeigt AWS OpsWorks Stacks die Seite „Ebenen“ an. Klicken Sie im Navigationsbereich auf Instanzen und dann unter PHP App Server auf Instanz hinzufügen.

2. Konfigurieren der Instance

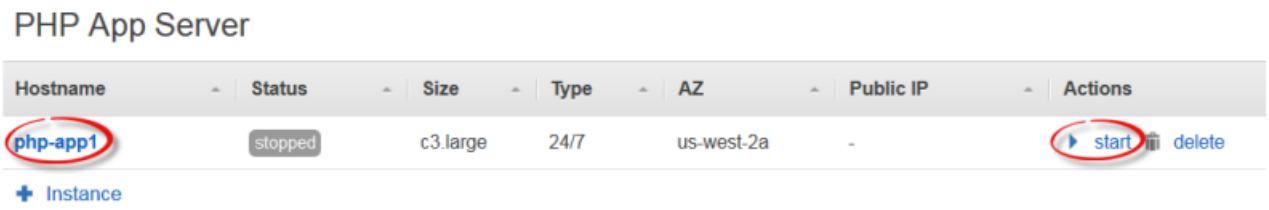
   Jede Instanz hat einen Standard-Hostnamen, der von AWS OpsWorks Stacks für Sie generiert wird. In diesem Beispiel fügt AWS OpsWorks Stacks dem Kurznamen der Ebene einfach eine Zahl hinzu. Sie können jede Instance getrennt konfigurieren, einschließlich der Übersteuerung einiger Standardeinstellungen, die Sie beim Erstellen des Stacks festgelegt haben, z. B. die Availability Zone oder das Betriebssystem. Akzeptieren Sie bei dieser Anleitung einfach die Standardeinstellungen und klicken Sie auf Add Instance (Instance hinzufügen), um der Ebene eine Instance hinzuzufügen. Weitere Informationen finden Sie unter Instances.



3. Starten der Instance

Bisher haben Sie nur die Konfiguration der Instance festgelegt. Sie müssen eine Instance starten, um eine laufende Amazon EC2 EC2-Instance zu erstellen. AWS OpsWorks Stacks verwendet dann die Konfigurationseinstellungen, um eine Amazon EC2 EC2-Instance in der angegebenen Availability Zone zu starten. Die Details, die beim Starten einer Instance zu berücksichtigen sind, hängen vom Skalierungstyp der Instance ab. Im vorherigen Schritt haben Sie eine Instance mit dem Standardskalierungstyp 24/7 erstellt, der manuell gestartet und so lange ausgeführt wird, bis Sie ihn manuell beenden. Sie können auch zeit- und lastbasierte Skalierungstypen erstellen, bei denen AWS OpsWorks Stacks automatisch auf der Grundlage eines Zeitplans oder der aktuellen Auslastung startet und stoppt. Weitere Informationen finden Sie unter Verwaltung der Last mit zeit- und lastbasierten Instanzen.

Gehen Sie zu php-app1 unter PHP App Server und klicken Sie in der Spalte Aktionen der Zeile auf Start, um die Instanz zu starten.

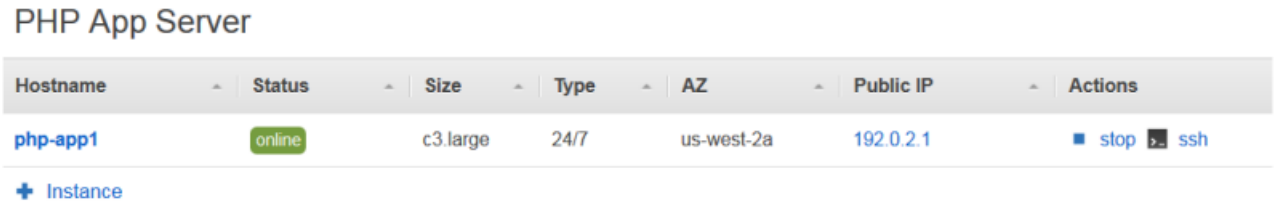


4. Überwachen des Instance-Status beim Start

   Normalerweise dauert es einige Minuten, um die Amazon EC2 EC2-Instance zu starten und die Pakete zu installieren. Während des Startprozesses zeigt das Feld Status der Instance folgende Werte an:

   1. angefordert — AWS OpsWorks Stacks hat den Amazon EC2-Service aufgerufen, um die Amazon EC2 EC2-Instance zu erstellen.
   2. ausstehend — AWS OpsWorks Stacks wartet auf den Start der Amazon EC2 EC2-Instance.
   3. booten — Die Amazon EC2 EC2-Instance bootet.
   4. running_setup — Der AWS OpsWorks Stacks-Agent führt die Setup-Rezepte des Layers aus, die Aufgaben wie das Konfigurieren und Installieren von Paketen übernehmen, und die Deploy-Rezepte, mit denen alle Apps auf der Instance bereitgestellt werden.
   5. online – Die Instance ist bereit zur Nutzung.

   Nachdem php-app1 online ist, sollte die Seite Instances (Instances) wie folgt aussehen:



   Die Seite beginnt mit einem Überblick über alle Instances Ihrer Stacks. Gegenwärtig wird eine Online-Instance angezeigt. Achten Sie in der php-app1-Spalte Actions (Aktionen) darauf, dass stop (Anhalten), wodurch die Instance gestoppt wird, start (Starten) und delete (Löschen) ersetzt hat.

Schritt 2.4: Erstellen und Bereitstellen einer Anwendung – Chef 11

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Um den Nutzen zu MyStack erhöhen, müssen Sie eine App auf der PHP App Server-Instanz bereitstellen. Sie speichern den Anwendungscode und die zugehörigen Dateien in einem Repository, wie Git. Sie müssen einige Schritte ausführen, um diese Dateien auf den Anwendungsserver zu übertragen:

> **Note**
>
> Die in diesem Abschnitt beschriebenen Schritte gelten für Chef 11-Stacks. Weitere Informationen zum Hinzufügen von Anwendungen zu Ebenen in Chef 12-Stacks finden Sie unter Hinzufügen von Apps.

1. Erstellen einer App

   Eine App enthält die Informationen, die AWS OpsWorks Stacks benötigt, um den Code und die zugehörigen Dateien aus dem Repository herunterzuladen. Sie können auch zusätzliche Informationen wie die Domäne der Anwendung festlegen.

2. Stellen Sie die Anwendung auf Ihren Anwendungsservern bereit.

   Wenn Sie eine App bereitstellen, löst AWS OpsWorks Stacks ein Deploy-Lifecycle-Ereignis aus. Der Agent führt anschließend die Funktion „Rezepte bereitstellen" der Instance aus, wodurch die Dateien in das richtige Verzeichnis heruntergeladen werden, zusammen mit den zugehörigen Aufgaben, wie Serverkonfiguration, Neustart des Dienstes etc.

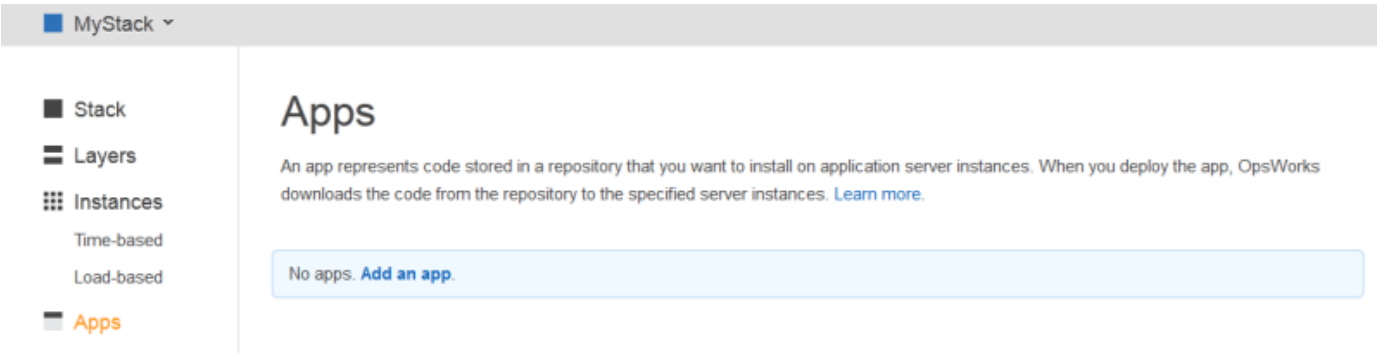
> **Note**
>
> Wenn Sie eine neue Instanz erstellen, stellt AWS OpsWorks Stacks automatisch alle vorhandenen Apps auf der Instanz bereit. Wenn Sie jedoch eine neue Anwendung erstellen oder eine bestehende aktualisieren, müssen Sie die Anwendung manuell bereitstellen oder um alle vorhandenen Instances aktualisieren.

In diesem Schritt wird gezeigt, wie Sie eine Beispiel-Anwendung von einem öffentlichen Git-Repository manuell auf einem Anwendungsserver bereitstellen. Wenn Sie die Anwendung untersuchen möchten, gehen Sie zu https://github.com/amazonwebservices/ opsworks-demo-php-simple -app. Die in diesem Beispiel verwendete Anwendung befindet sich im Zweig Version1. AWS OpsWorks Stacks unterstützt auch mehrere andere Repository-Typen. Weitere Informationen finden Sie unter Anwendungsquelle.

Erstellen und Bereitstellen einer Anwendung

1. Öffnen der Seite „Anwendungen“

   Klicken Sie im Navigationsbereich auf Apps (Anwendungen) und klicken Sie auf der Seite Apps (Anwendungen) auf Add an app (Anwendung hinzufügen).



2. Konfigurieren der Anwendung

   Geben Sie auf der Seite App (Anwendung) die folgenden Werte an:

   Name

   Der Name der App, den AWS OpsWorks Stacks für Anzeigezwecke verwendet. Die Beispiel-App trägt den Namen**`SimplePHPApp`**. AWS OpsWorks Stacks generiert außerdem einen Kurznamen — in diesem Beispiel simplephpapp —, der intern und in den Deploy-Rezepten verwendet wird, wie später beschrieben.

   Typ

   Der Anwendungstyp, über den festgelegt wird, wo die Anwendung bereitgestellt wird. Das Beispiel verwendet PHP, das die App auf PHP App Server-Instanzen bereitstellt.

Datenquellentyp

Ein zugehöriger Datenbankserver. Wählen Sie jetzt None (Keine Angabe) aus; eine Einführung in Datenbankserver finden Sie unter Schritt 3: Hinzufügen eines Backend-Datenspeichers.

Repository-Typ

Der Repository-Typ der Anwendung. Die Beispielanwendung ist in einem Git-Repository gespeichert.

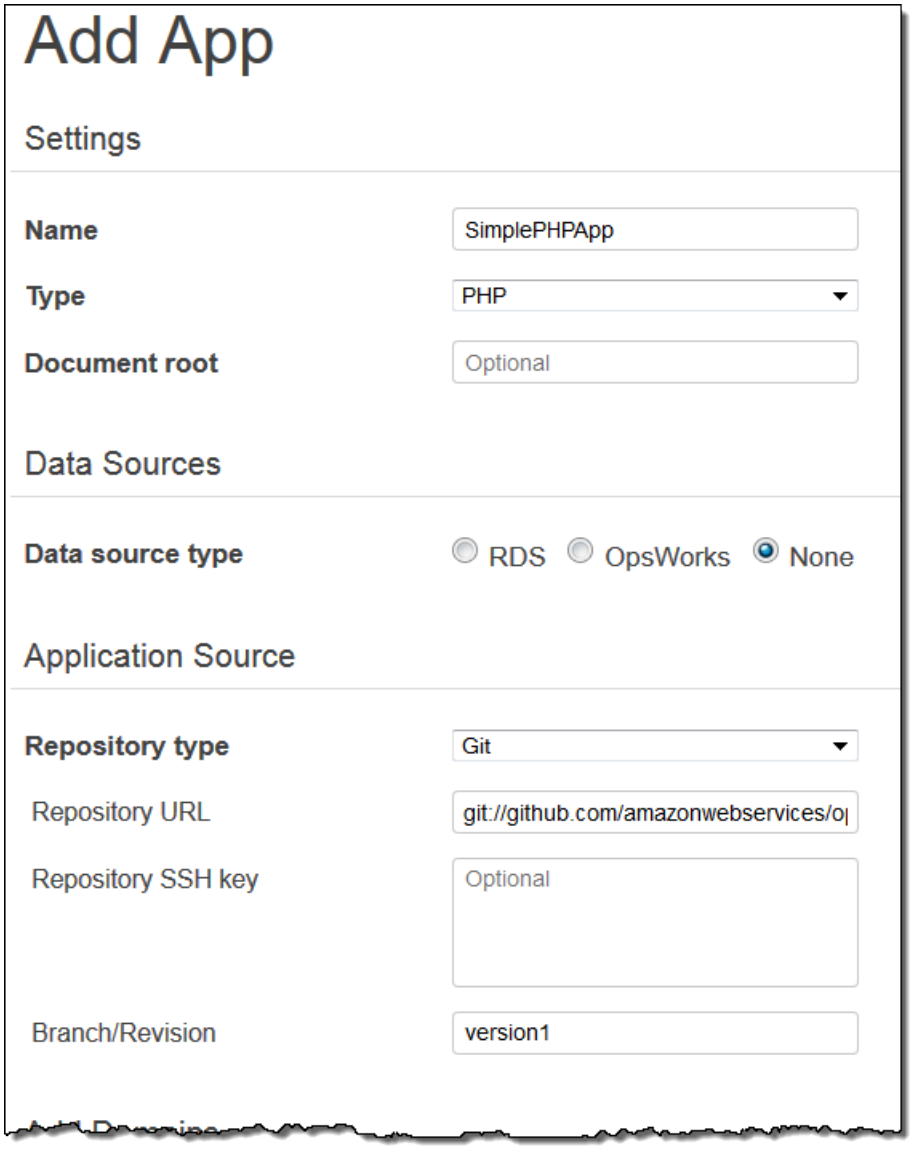
Repository-URL

Die Repository-URL der Anwendung. Die Beispiel-URL lautet: **`git://github.com/awslabs/opsworks-demo-php-simple-app.git`**

Branch/Revision

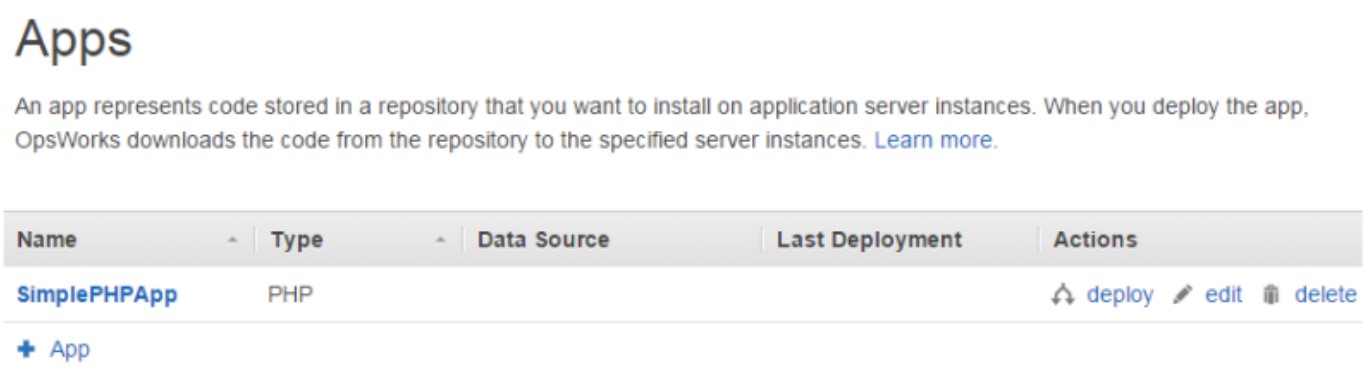
Die Branch oder Version der Anwendung. Dieser Teil der Anleitung verwendet den Zweig **`version1`**.

Behalten Sie die Standardwerte für die verbleibenden Einstellungen bei und klicken Sie auf Add App (Anwendung hinzufügen). Weitere Informationen finden Sie unter Hinzufügen von Apps.



3. Öffnen der Seite „Bereitstellung“

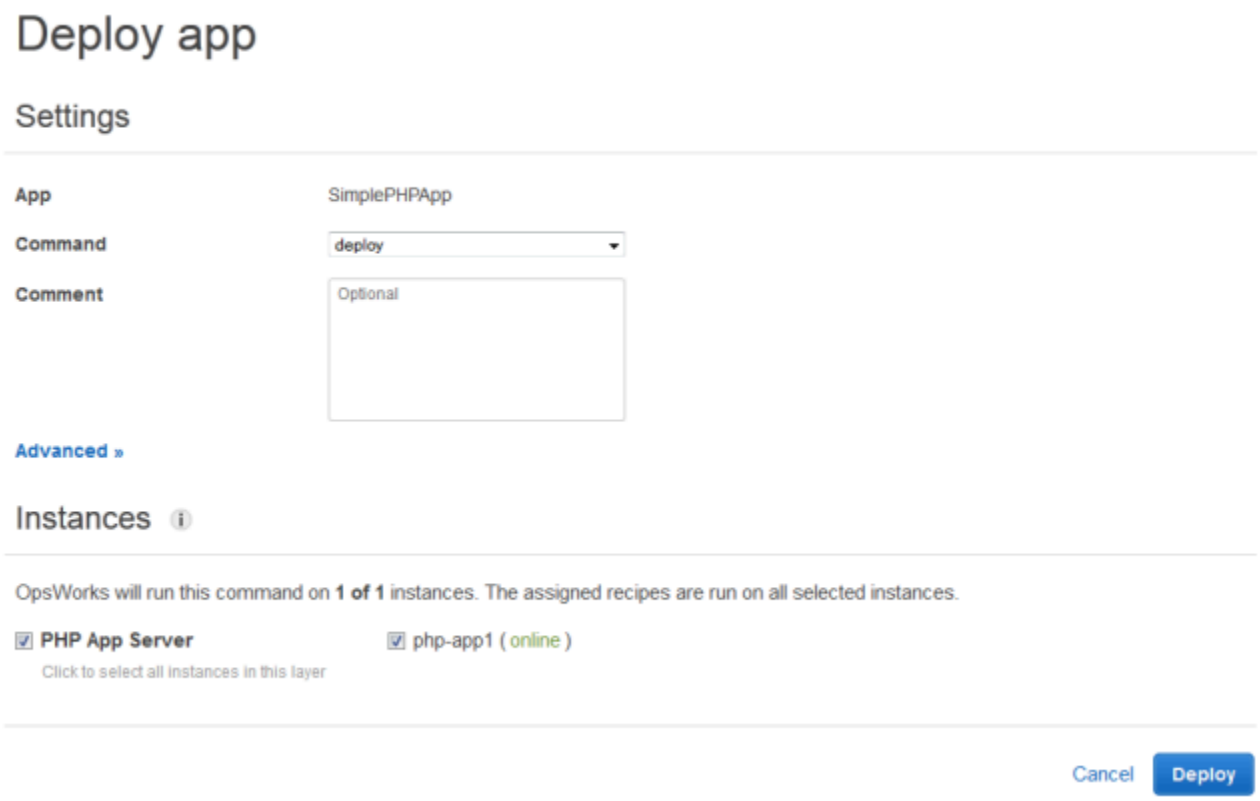
   Zum Installieren des Programmcodes auf dem Server müssen Sie die Anwendung bereitstellen. Klicken Sie dazu auf deploy (Bereitstellen) in der SimplePHPApp-Spalte Actions (Aktionen).



4. Bereitstellen der Anwendung

   Wenn Sie eine App bereitstellen, führt der Agent die Deploy-Rezepte auf der PHP App Server-Instanz aus, wodurch die Anwendung heruntergeladen und konfiguriert wird.

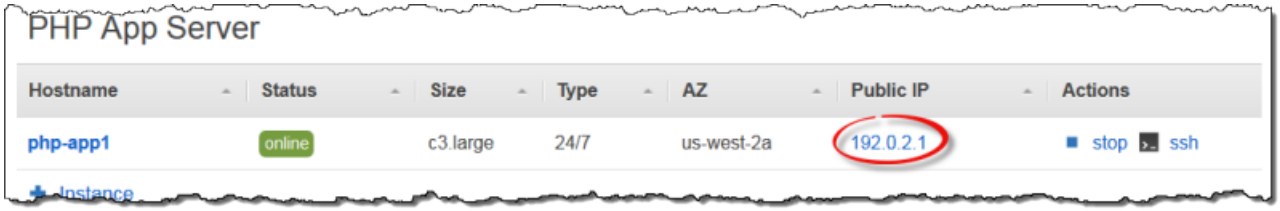
   Command (Befehl) muss bereits auf deploy (Bereitstellen) festgelegt sein. Übernehmen Sie die Standardeinstellungen für die anderen Einstellungen und klicken Sie auf Deploy (Bereitstellen), um die Anwendung bereitzustellen.



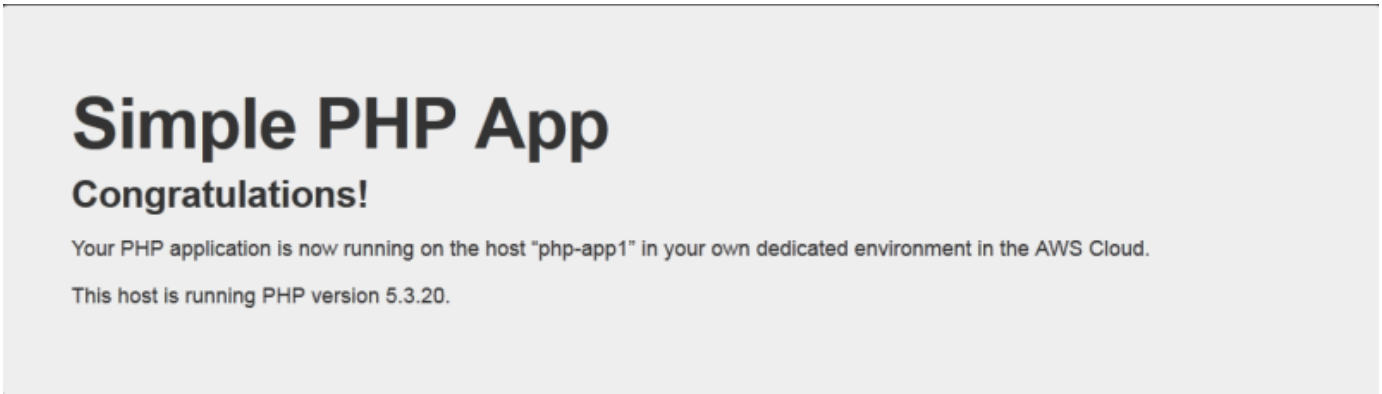
Wenn die Bereitstellung abgeschlossen ist, zeigt die Seite Deployment (Bereitstellung) den Status Successful (Erfolgreich) an und neben php-app1 ein grünes Häkchen.

5. Ausführen von SimplePHPApp

SimplePHPApp ist jetzt installiert und einsatzbereit. Klicken Sie zum Ausführen im Navigationsbereich auf Instances (Instances), um zur Seite Instances (Instances) zu gelangen. Klicken Sie dann auf die öffentliche IP-Adresse der php-app1 Instance.



In Ihrem Browser sollte sich eine Seite wie die folgende öffnen.

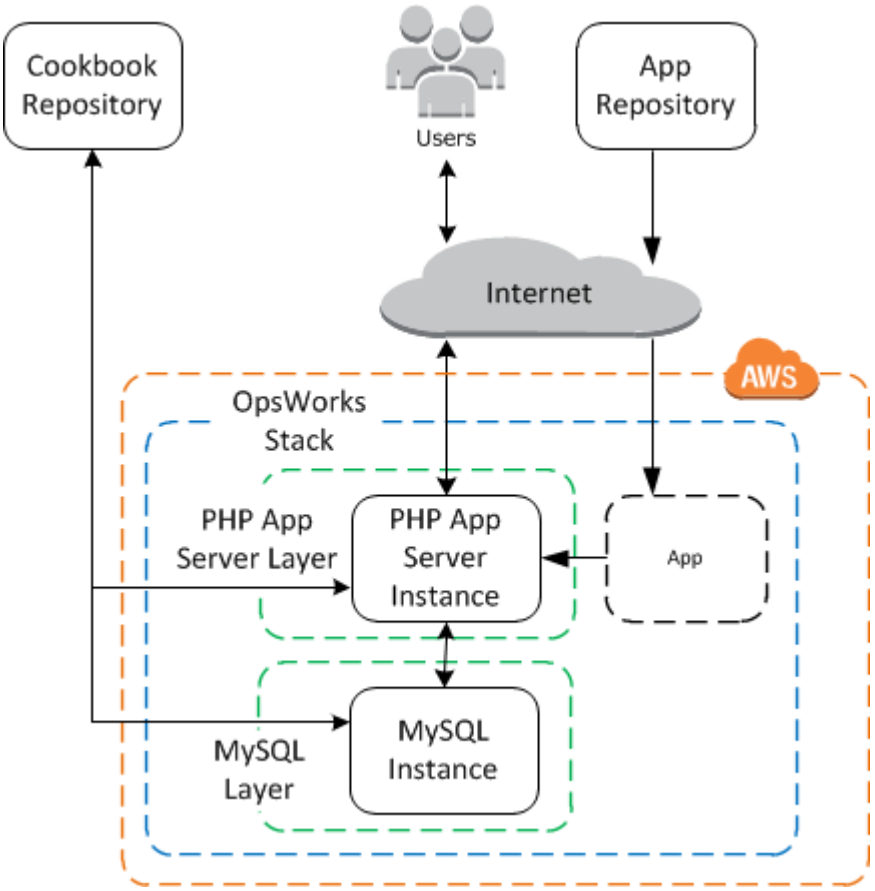


> **Note**
>
> In dieser Anleitung wird davon ausgegangen, dass Sie mit dem nächsten Abschnitt fortfahren und die gesamte Anleitung in einer Sitzung abschließen. Wenn Sie möchten, können Sie jederzeit aufhören und später weitermachen, indem Sie sich bei AWS OpsWorks Stacks anmelden und den Stack öffnen. Für verwendete AWS-Ressourcen, wie z. B. Online-Instances, werden jedoch Gebühren erhoben. Um unnötige Kosten zu vermeiden, können Sie Ihre Instance anhalten, wodurch die zugehörige EC2 Instance beendet wird. Sie können die Instances erneut starten, wenn Sie fortfahren möchten.

Schritt 3: Hinzufügen eines Backend-Datenspeichers

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Unter Schritt 2.1: Erstellen eines Stacks – Chef 11 wurde die Erstellung eines Stacks erläutert, der eine PHP-Anwendung verarbeitete. Allerdings war dies eine sehr einfache Anwendung ohne viele Funktionen, in der nur statischer Text angezeigt wurde. In Produktionsanwendungen wird häufig ein Backend-Datenspeicher verwendet. Dies ergibt eine Stack-Konfiguration, die in etwa wie in der folgenden Abbildung dargestellt aussieht.



In diesem Abschnitt wird gezeigt, wie Sie die Erweiterung MyStack um einen Backend-MySQL-Datenbankserver erweitern können. Sie müssen jedoch mehrere Aufgaben ausführen, als einfach

nur einen MySQL-Server zum Stack hinzufügen. Sie müssen die App auch so konfigurieren, dass sie ordnungsgemäß mit dem Datenbankserver kommuniziert. AWS OpsWorks Stacks erledigt das nicht für Sie. Sie müssen einige benutzerdefinierte Rezepte implementieren, um diese Aufgabe zu bewältigen.

Themen

Schritt 3.1: Hinzufügen einer Backend-Datenbank

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Die neue Version von SimplePhpApp speichert ihre Daten in einer Backend-Datenbank. AWS OpsWorks Stacks unterstützt zwei Arten von Datenbankservern:

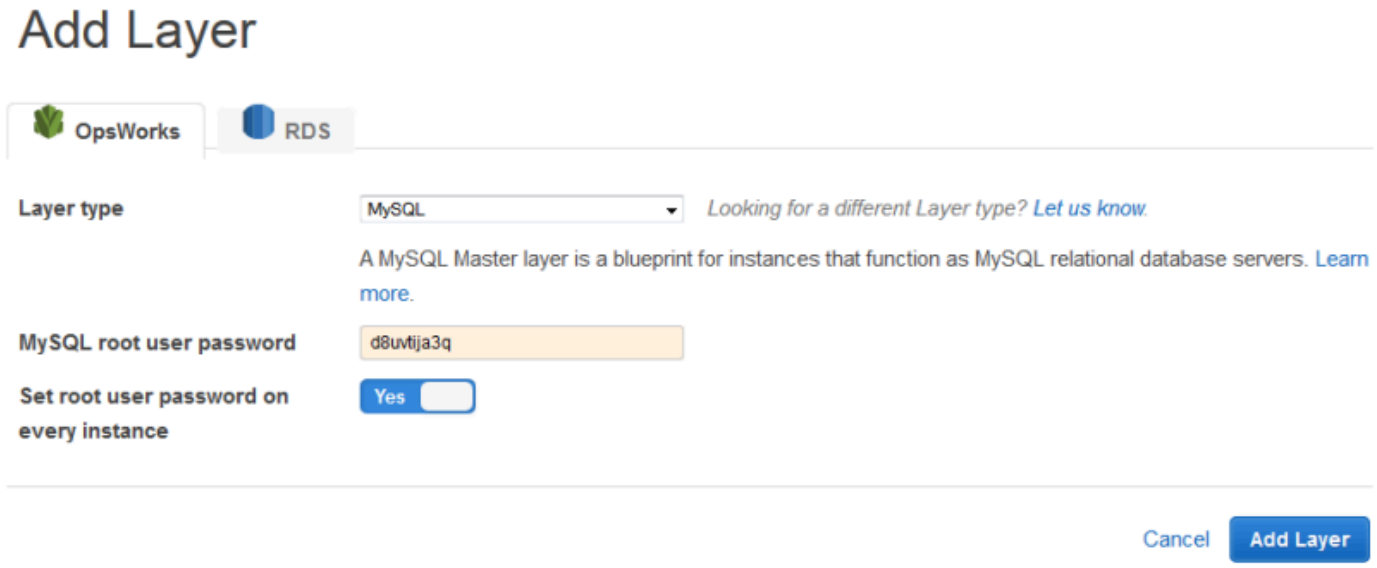
- Die MySQL AWS OpsWorks Stacks-Ebene ist eine Blaupause für die Erstellung von Amazon EC2 EC2-Instances, die einen MySQL-Datenbankmaster hosten.
- Die Amazon RDS-Serviceschicht bietet eine Möglichkeit, eine Amazon RDS-Instance in einen Stack zu integrieren.

Sie können auch andere Datenbanken wie Amazon DynamoDB verwenden oder eine benutzerdefinierte Ebene erstellen, um Datenbanken wie MongoDB zu unterstützen. Weitere Informationen finden Sie unter the section called “Verwenden eines Backend-Datenspeichers”.

In diesem Beispiel wird eine MySQL-Schicht verwendet.

Um eine MySQL-Ebene hinzuzufügen MyStack

1. Klicken Sie auf der Seite Layers (Ebenen) auf + Layer (+ Ebene).
2. Wählen Sie auf der Seite Add Layer (Ebene hinzufügen) für Layer type (Typ der Ebene) die Option MySQL aus, übernehmen Sie die Standardeinstellungen und klicken Sie auf Add Layer (Ebene hinzufügen).



Um eine Instanz zur MySQL-Ebene hinzuzufügen

1. Klicken Sie auf der Seite Layers (Ebenen) in der Zeile MySQL auf Add an instance (Instance hinzufügen).
2. Klicken Sie auf der Seite Instances unter MySQL auf Add an instance (Instance hinzufügen).
3. Akzeptieren Sie die Standardwerte und klicken Sie auf Add instance (Instance hinzufügen), aber starten Sie sie noch nicht.

> **Note**
>
> AWS OpsWorks Stacks erstellt automatisch eine Datenbank, die nach dem Kurznamen der App benannt wird, in diesem Beispiel simplephpapp. Sie benötigen diesen Namen, wenn Sie Chef-Rezepte für die Interaktion mit der Datenbank verwenden möchten.

Schritt 3.2: Aktualisieren von SimplePHPApp

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Zunächst benötigen Sie eine neue Version von SimplePHPApp, in der ein Backend-Datenspeicher verwendet wird. Mit AWS OpsWorks Stacks ist die Aktualisierung einer Anwendung ganz einfach. Wenn Sie ein Git- oder Subversion-Repository verwenden, kann jede Anwendungsversion über einen separaten Repository-Branch verfügen. Die Beispielanwendung speichert eine Version der Anwendung, die eine Backend-Datenbank im version2-Branch des Git-Repositorys verwendet. Sie müssen einfach die Konfiguration der Anwendung aktualisieren, um den neuen Branch anzugeben, und die Anwendung erneut bereitstellen.

So aktualisieren Sie SimplePHPApp

1. Öffnen Sie die Bearbeitungsseite der Anwendung.

   Klicken Sie im Navigationsbereich auf Apps (Anwendungen) und anschließend in der Spalte Actions (Aktionen) der Zeile SimplePHPApp auf edit (Bearbeiten).

2. Aktualisieren Sie die Konfiguration der Anwendung.

   Ändern Sie die folgenden Einstellungen.

Branch/Revision

Diese Einstellung gibt den Repository-Branch der Anwendung an. Mit der ersten Version von SimplePHPApp konnte keine Verbindung mit einer Datenbank hergestellt werden. Um eine datenbankfähige Version der Anwendung zu verwenden, stellen Sie diesen Wert auf **`version2`** ein.

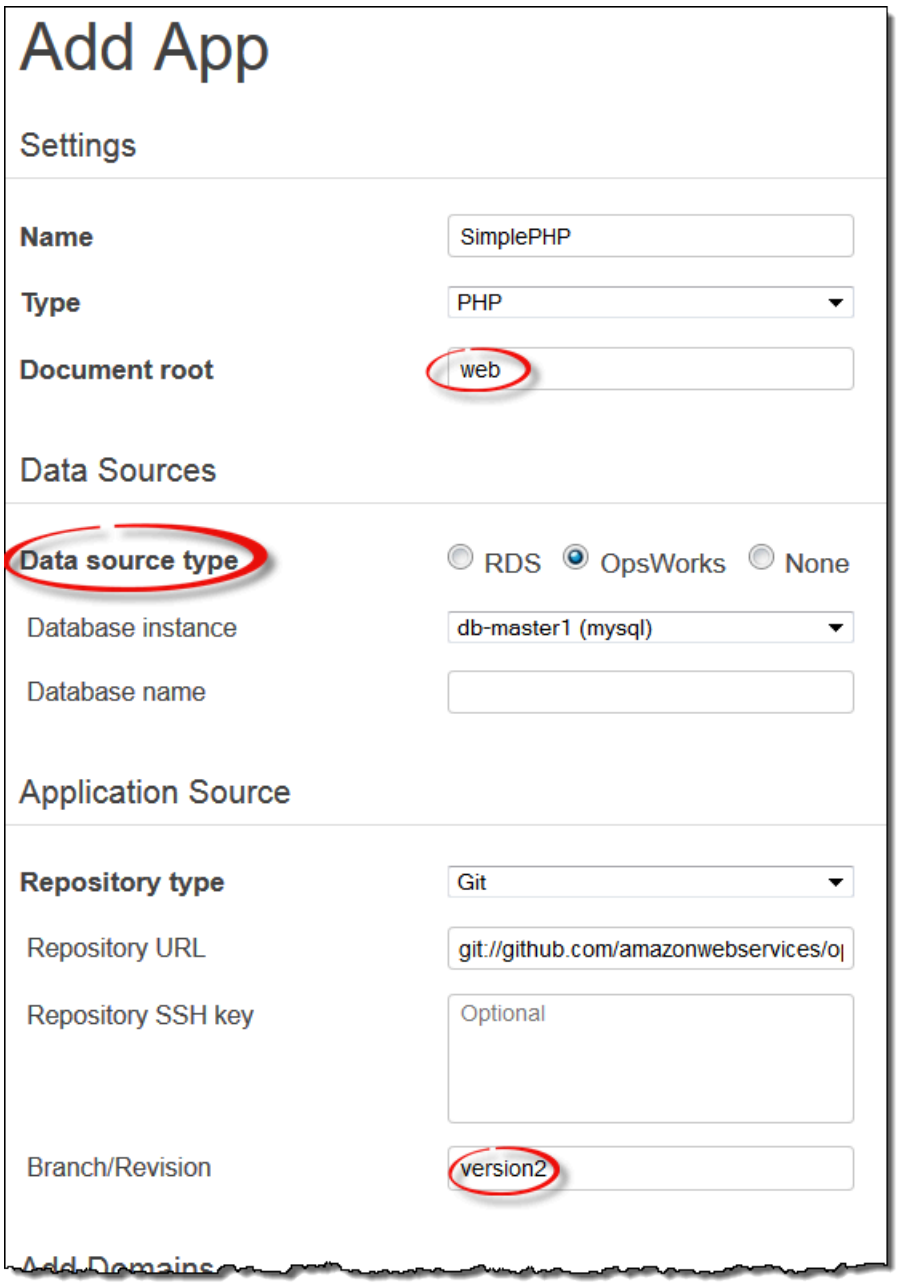
Document root (Basisverzeichnis)

Diese Einstellung gibt den Stammordner Ihrer Anwendung an. In der ersten Version von SimplePHPApp wurde die Standardeinstellung verwendet, mit der die Datei `index.php` im Standardstammordner des Servers installiert wird (`/srv/www` für PHP-Anwendungen). Wenn Sie hier einen Unterordner angeben — nur den Namen, kein vorangestelltes „/“ — hängt AWS OpsWorks Stacks ihn an den Standardordnerpfad an. Version 2 von SimplePHPApp sollte im Ordner `/srv/www/web` platziert werden. Legen Sie daher für die Einstellung Document root (Basisverzeichnis) den Wert **`web`** fest.

Data source type (Datenquellentyp)

Mit dieser Einstellung wird ein Datenbankserver mit der Anwendung verknüpft. Das Beispiel verwendet die MySQL-Instanz, die Sie im vorherigen Schritt erstellt haben. Setzen Sie also Datenquellentyp auf OpsWorks und Datenbankinstanz auf die Instanz, die Sie im vorherigen Schritt erstellt haben, db-master1 (mysql). Lassen Sie den Datenbanknamen leer. AWS OpsWorks Stacks erstellt eine Datenbank auf dem Server mit dem Kurznamen der App, simplephpapp.

Klicken Sie zum Speichern der neuen Konfiguration dann auf Save (Speichern).



3. Starten Sie die MySQL-Instanz.

Nachdem Sie eine App aktualisiert haben, stellt AWS OpsWorks Stacks die neue App-Version automatisch auf allen neuen App-Server-Instanzen bereit, wenn Sie sie starten. AWS OpsWorks Stacks stellt die neue App-Version jedoch nicht automatisch auf vorhandenen Serverinstanzen bereit. Sie müssen dies manuell tun, wie unter beschrieben. Schritt 2.4: Erstellen und Bereitstellen einer Anwendung – Chef 11 Sie können nun die aktualisierte SimplePHPApp-Anwendung bereitstellen. Für dieses Beispiel empfiehlt es sich jedoch zu warten.

Ein kurzer Exkurs: Kochbücher, Rezepte und Stacks-Attribute AWS OpsWorks

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie verfügen nun über Anwendungs- und Datenbankserver, die allerdings noch nicht ganz einsatzbereit sind. Du musst noch die Datenbank einrichten und die Verbindungseinstellungen der App konfigurieren. AWS OpsWorks Stacks erledigt diese Aufgaben nicht automatisch, unterstützt aber Chef-Kochbücher, Rezepte und dynamische Attribute. Sie können zwei Rezepte implementieren, eines zum Einrichten der Datenbank und eines zum Konfigurieren der Verbindungseinstellungen der App, und AWS OpsWorks Stacks diese für Sie ausführen lassen.

Das phpapp-Rezeptbuch, das die erforderlichen Rezepte enthält, ist bereits implementiert und einsatzbereit. Gegebenenfalls können Sie einfach zu Schritt 3.3: Fügen Sie die benutzerdefinierten Kochbücher hinzu MyStack wechseln. Wenn Sie mehr erfahren möchten, finden Sie in diesem Abschnitt einige Hintergrundinformationen zu Rezeptbüchern und Rezepten sowie eine Beschreibung der Funktionsweise von Rezepten. Zum Anzeigen des Rezeptbuchs rufen Sie die Website phpapp cookbook auf.

Themen

- Rezepte und Attribute
- Einrichten der Datenbank
- Verknüpfen der Anwendung mit der Datenbank

Rezepte und Attribute

Bei einem Chef-Rezept handelt es sich im Prinzip um eine spezialisierte Ruby-Anwendung, die Aufgaben auf einer Instance ausführt, z. B. Installieren von Paketen, Erstellen von Konfigurationsdateien, Ausführen von Shell-Befehlen usw. Gruppen zusammengehöriger Rezepte sind in Rezeptbücher zusammengefasst, die auch unterstützende Dateien enthalten, wie z. B. Vorlagen zur Erstellung von Konfigurationsdateien.

AWS OpsWorks Stacks verfügt über eine Reihe von Kochbüchern, die die integrierten Ebenen unterstützen. Darüber hinaus können Sie benutzerdefinierte Rezeptbücher mit Ihren eigenen Rezepten erstellen, um benutzerdefinierte Aufgaben auf Ihren Instances auszuführen. In diesem Thema finden Sie eine kurze Einführung in Rezepte sowie eine Beschreibung dazu, wie Sie mit diesen die Datenbank einrichten und die Verbindungseinstellungen der Anwendung konfigurieren können. Weitere Informationen zu Rezeptbüchern und Rezepten finden Sie unter Cookbooks und Rezepte oder Stacks anpassen AWS OpsWorks.

Rezepte hängen bezüglich Eingabedaten normalerweise von Chef-Attributen ab:

- Einige dieser Attribute werden durch Chef definiert und bieten grundlegende Informationen zur Instance, wie beispielsweise das Betriebssystem.
- AWS OpsWorks Stacks definiert eine Reihe von Attributen, die Informationen über den Stack — wie die Layer-Konfigurationen — und über bereitgestellte Apps — wie das App-Repository — enthalten.

  Sie können benutzerdefinierte Attribute zu dieser Gruppe hinzufügen, indem Sie dem Stack oder der Bereitstellung eine benutzerdefinierte JSON-Datei zuweisen.
- Ihre Rezeptbücher können auch Attribute definieren, die für das Rezeptbuch spezifisch sind.

  Die phpapp-Rezeptbuch-Attribute sind in der Datei `attributes/default.rb` definiert.

Eine vollständige Liste der Stacks-Attribute finden Sie unter und. AWS OpsWorks Stack-Konfigurations- und Bereitstellungsattribute: Linux Integrierte Rezeptbuchattribute Weitere Informationen finden Sie unter Überschreiben der Attribute.

Attribute sind in einer hierarchischen Struktur organisiert, die als JSON-Objekt dargestellt werden kann.

Sie integrieren diese Daten mithilfe der Chef-Knotensyntax in Ihrer Anwendung, wie beispielsweise die folgende:

```
[:deploy][:simplephpapp][:database][:username]
```

Der Knoten `deploy` hat einen einzelnen Anwendungsknoten, `simplephpapp`, der Informationen zur Datenbank der Anwendung, zum Git-Repository usw. enthält. Im Beispiel ist der Wert des Benutzernamens für die Datenbank dargestellt, der in `root` aufgelöst wird.

Einrichten der Datenbank

Die in der MySQL-Schicht integrierten Setup-Rezepte erstellen automatisch eine Datenbank für die App, die mit dem Kurznamen der App benannt ist. In diesem Beispiel haben Sie also bereits eine Datenbank mit dem Namen simplephpapp. Sie müssen jedoch die Einrichtung abschließen, indem Sie eine Tabelle für die Anwendung zum Speichern ihrer Daten erstellen. Sie könnten die Tabelle manuell erstellen, aber ein besserer Ansatz besteht darin, ein benutzerdefiniertes Rezept für die Bearbeitung der Aufgabe zu implementieren und AWS OpsWorks Stacks es für Sie ausführen zu lassen. In diesem Abschnitt wird die Implementierung des Rezepts `dbsetup.rb` erläutert. Das Verfahren, mit dem AWS OpsWorks Stacks das Rezept ausführen lässt, wird später beschrieben.

Um das Rezept im Repository anzuzeigen, rufen Sie dbsetup.rb auf. Im folgenden Beispiel ist das Rezept `dbsetup.rb` dargestellt.

Bei `execute` handelt es sich um eine Chef-Ressource, die einen angegebenen Befehl ausführt. In diesem Fall ist das ein MySQL-Befehl, mit dem eine Tabelle erstellt wird. Die Richtlinie `not_if` sorgt dafür, dass der Befehl nicht ausgeführt wird, wenn die angegebene Tabelle bereits vorhanden ist. Weitere Informationen zu Chef-Ressourcen finden Sie auf der Website About Resources and Providers.

Das Rezept fügt Attributwerte mithilfe der vorher erläuterten Knotensyntax in die Befehlszeichenfolge ein. Beispielsweise wird mit der folgenden Syntax der Benutzername der Datenbank eingefügt.

```
#{deploy[:database][:username]}
```

Entpacken wir nun diesen leicht kryptischen Code:

- Bei jeder Iteration ist `deploy` auf den aktuellen Anwendungsknoten eingestellt, sodass er in `[:deploy][:app_name]` aufgelöst wird. In diesem Beispiel wird er in `[:deploy][:simplephpapp]` aufgelöst.
- Mithilfe der vorher genannten Attributwerte der Bereitstellung wird der gesamte Knoten in `root` aufgelöst.
- Sie umschließen den Knoten in #{ }, um ihn in eine Zeichenfolge einzufügen.

Die meisten anderen Knoten werden auf ähnliche Weise aufgelöst. Die Ausnahme stellt der Knoten `#{node[:phpapp][:dbtable]}` dar, der durch die Attributdatei des benutzerdefinierten Rezeptbuchs definiert ist und in den Tabellennamen `urler` aufgelöst wird. Der eigentliche Befehl, der auf der MySQL-Instanz ausgeführt wird, lautet daher:

```
"/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))'
"
```

Mit diesem Befehl wird unter Verwendung der Anmeldeinformationen und des Datenbanknamens aus den Bereitstellungsattributen die Tabelle `urler` mit ID, Autor und Nachrichtenfeldern erstellt.

Verknüpfen der Anwendung mit der Datenbank

Die zweite wichtige Komponente ist die Anwendung, die Verbindungsinformationen wie das Datenbankpasswort für den Zugriff auf die Tabelle benötigt. SimplePHPApp hat effektiv nur eine Arbeitsdatei, `app.php`. Die einzige Aufgabe von `index.php` besteht darin, die Datei `app.php` zu laden.

`app.php` enthält die Datei `db-connect.php`, die die Datenbankverbindung verarbeitet, jedoch befindet sich die Datei nicht im Repository. Sie können die Datei `db-connect.php` nicht im Voraus erstellen, da sie die Datenbank basierend auf der bestimmten Instance definiert. Stattdessen generiert das Rezept `appsetup.rb` die Datei `db-connect.php` mithilfe der Verbindungsdaten aus den Bereitstellungsattributen.

Um das Rezept im Repository anzuzeigen, rufen Sie appsetup.rb auf. Im folgenden Beispiel ist das Rezept `appsetup.rb` dargestellt.

`dbsetup.rbappsetup.rb`lteriert quasi über Apps im `deploy` Knoten — einfach nochmal phpapp —. Dieses Rezept führt einen Codeblock mit einer `script`-Ressource und einer `template`-Ressource aus.

Die `script` Ressource installiert Composer — einen Abhängigkeitsmanager für PHP-Anwendungen. Anschließend wird der Composer-Befehl `install` ausgeführt, um die Abhängigkeiten für die Beispielanwendung im Stammverzeichnis der Anwendung zu installieren.

Mit der Ressource `template` wird die Datei `db-connect.php` erstellt und im Verzeichnis `/srv/www/simplephpapp/current` gespeichert. Beachten Sie Folgendes:

- Das Rezept verwendet eine bedingte Anweisung zum Festlegen des Dateieigentümers, die vom Betriebssystem der Instance abhängt.
- Die Richtlinie `only_if` weist Chef an, die Vorlage nur dann zu generieren, wenn das angegebene Verzeichnis vorhanden ist.

Eine `template`-Ressource arbeitet mit einer Vorlage, die im Wesentlichen den gleichen Inhalt und die gleiche Struktur wie die zugehörige Datei hat, jedoch Platzhalter für verschiedene Datenwerte enthält. Der Parameter `source` gibt die Vorlage `db-connect.php.erb` an, bei der es sich um das Verzeichnis `templates/default` des phpapp-Rezeptbuchs handelt. Er enthält folgende Werte:

Bei Verarbeitung der Vorlage durch Chef werden die Platzhalter `<%= =>` in der Vorlagenressource durch den Wert der entsprechenden Variablen ersetzt, die wiederum aus den Bereitstellungsattributen stammen. Die generierte Datei sieht daher wie folgt aus:

Schritt 3.3: Fügen Sie die benutzerdefinierten Kochbücher hinzu MyStack

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Genauso wie Anwendungen speichern Sie benutzerdefinierte Rezeptbücher in einem Repository. Jeder Stack kann ein Repository mit einer Reihe von benutzerdefinierten Rezeptbüchern enthalten. Anschließend weisen Sie AWS OpsWorks Stacks an, Ihre benutzerdefinierten Kochbücher auf den Instanzen des Stacks zu installieren.

1. Klicken Sie im Navigationsbereich auf Stack (Stack), um die Seite für den aktuellen Stack anzuzeigen.
2. Klicken Sie auf Stack Settings (Stack-Einstellungen) und dann auf Edit (Bearbeiten).
3. Ändern Sie die Stack-Konfiguration wie folgt:
   - Verwenden Sie benutzerdefinierte Chef-Kochbücher — Ja
   - Repository-Typ — Git

- Repository-URL — **`git://github.com/amazonwebservices/opsworks-example-cookbooks.git`**

4. Klicken Sie zum Aktualisieren der Stack-Konfiguration auf Save (Speichern).

AWS OpsWorks Stacks installiert dann den Inhalt Ihres Kochbuch-Repositorys auf allen Instanzen des Stacks. Wenn Sie neue Instanzen erstellen, installiert AWS OpsWorks Stacks automatisch das Cookbook-Repository.

> **Note**
>
> Wenn du eines deiner Kochbücher aktualisieren oder neue Kochbücher zum Repository hinzufügen musst, kannst du das tun, ohne die Stack-Einstellungen zu ändern. AWS OpsWorks Stacks installiert die aktualisierten Kochbücher automatisch auf allen neuen Instanzen. AWS OpsWorks Stacks installiert jedoch nicht automatisch aktualisierte Kochbücher auf den Online-Instanzen des Stacks. Sie müssen AWS OpsWorks Stacks explizit anweisen, die Kochbücher zu aktualisieren, indem Sie den Stack-Befehl ausführen. `Update Cookbooks` Weitere Informationen finden Sie unter Ausführen von Stack-Befehlen.

Schritt 3.4: Ausführen der Rezepte

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Nachdem Sie Ihr benutzerdefiniertes Rezeptbuch erstellt haben, müssen Sie die Rezepte auf den entsprechenden Instances ausführen. Sie können sie aber auch manuell ausführen. Rezepte müssen jedoch normalerweise zu planbaren Zeitpunkten im Lebenszyklus einer Instance ausgeführt werden, z. B. nach dem Start der Instance oder bei der Bereitstellung einer Anwendung. In diesem Abschnitt wird ein viel einfacherer Ansatz beschrieben: Lass AWS OpsWorks Stacks sie automatisch zum richtigen Zeitpunkt für dich ausführen.

AWS OpsWorks Stacks unterstützt eine Reihe von Lebenszyklusereignissen, die das Ausführen von Rezepten vereinfachen. Beispielsweise erfolgt das Setup-Ereignis nach dem Hochfahren einer Instance und das Bereitstellungsereignis bei der Bereitstellung einer Anwendung. Jeder Layer verfügt über eine Reihe von integrierten Rezepten, die mit dem jeweiligen Lebenszyklusereignis verknüpft sind. Wenn ein Lebenszyklusereignis auf einer Instance auftritt, führt der Agent die zugehörigen Rezepte für den jeweiligen Instance-Layer aus. Damit AWS OpsWorks Stacks ein benutzerdefiniertes Rezept automatisch ausführt, fügen Sie es dem entsprechenden Lebenszyklusereignis auf der entsprechenden Ebene hinzu. Der Agent führt das Rezept dann aus, wenn die integrierten Rezepte fertig sind.

Für dieses Beispiel müssen Sie zwei Rezepte ausführen, `dbsetup.rb` auf der MySQLInstance und `appsetup.rb` auf der PHP App Server-Instanz.

> **Note**
>
> Sie legen die Rezepte in der Konsole im Format *`cookbook_name`*::*`recipe_name`* fest, wobei *`recipe_name`* nicht die Erweiterung ".rb" enthält. Zum Beispiel verweisen Sie auf `dbsetup.rb` als **`phpapp::dbsetup`**.

So weisen Sie benutzerdefinierte Rezepte zu Lebenszyklusereignissen hinzu

1. Klicken Sie auf der Seite Layers für MySQL auf Rezepte und dann auf Bearbeiten.
2. Geben Sie im Abschnitt Custom Chef recipes (Benutzerdefinierte Chef-Rezepte) das Rezept **`phpapp::dbsetup`** für Deploy (Bereitstellen) ein.

3. Klicken Sie auf das +-Symbol, um das Rezept dem Ereignis zuzuweisen, und klicken Sie dann auf Save (Speichern), um die neue Ebenenkonfiguration zu speichern.
4. Kehren Sie zur Seite „Ebenen“ zurück und wiederholen Sie das Verfahren, **`phpapp::appsetup`** um das Deploy-Ereignis des PHP App Server-Layers zuzuweisen.

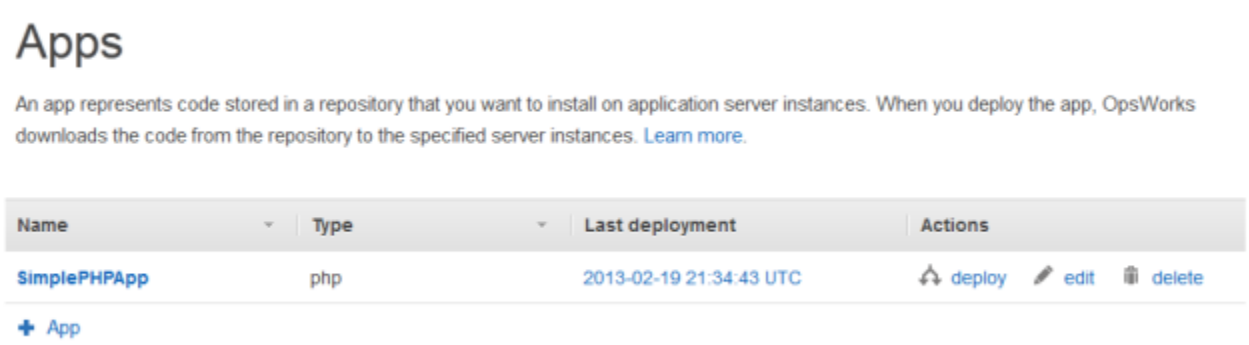
Schritt 3.5: Bereitstellen von SimplePHPApp, Version 2

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

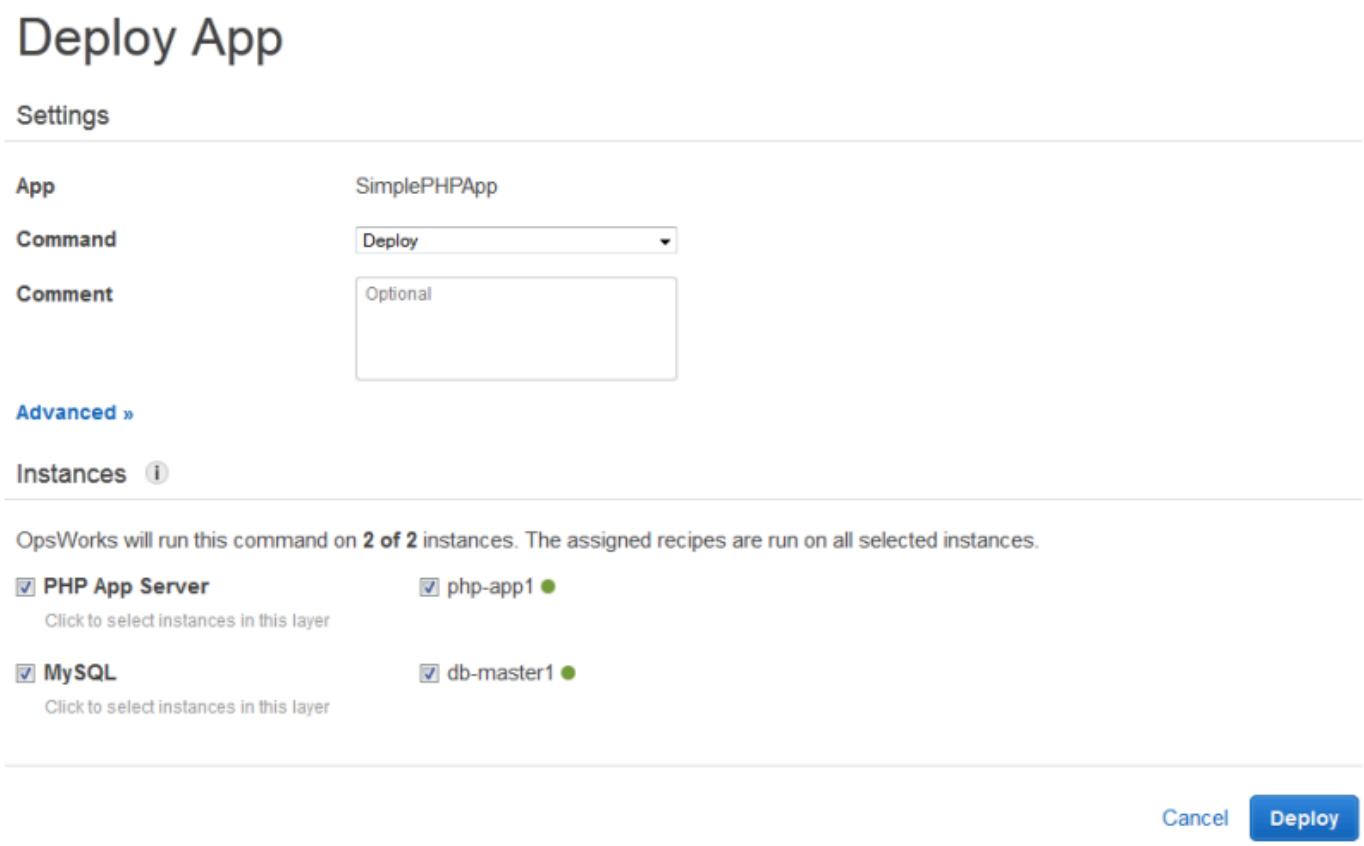
Der letzte Schritt besteht in der Bereitstellung der neuen Version von SimplePHPApp.

So stellen Sie SimplePHPApp bereit

1. Klicken Sie auf der Seite Apps auf deploy unter Actions in der App SimplePHPApp.



2. Übernehmen Sie die Standardeinstellungen und klicken Sie auf Deploy (Bereitstellen).



Durch Klicken auf Deploy (Bereitstellen) auf der Seite Deploy App (Anwendung bereitstellen) lösen Sie ein Bereitstellungs-Lebenszyklusereignis aus, durch das die Agenten zur Ausführung ihrer Bereitstellungsrezepte angewiesen werden. Standardmäßig wird das Ereignis auf allen Stack-Instances ausgelöst. Die integrierten Deploy-Rezepte stellen die App nur auf den entsprechenden Instanzen für den App-Typ bereit, in diesem Fall auf PHP-App-Server-Instanzen. Es ist jedoch häufig sinnvoll, das Bereitstellungsereignis auf anderen Instances auszulösen, damit sie auf Anwendungsbereitstellung reagieren können. In diesem Fall möchten Sie Deploy auch auf der MySQL-Instanz auslösen, um die Datenbank einzurichten.

Beachten Sie Folgendes:

- Der Agent auf der PHP App Server-Instanz führt das integrierte Rezept des Layers aus, gefolgt von `appsetup.rb`, das die Datenbankverbindung der App konfiguriert.
- Der Agent auf der MySQL-Instanz installiert nichts, aber er wird ausgeführt, `dbsetup.rb` um die URL-Tabelle zu erstellen.

Wenn die Bereitstellung abgeschlossen ist, ändert sich der Status auf der Seite Deployment (Bereitstellung) in successful (Erfolgreich).

Schritt 3.6: Ausführen von SimplePHPApp

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Nachdem für die Bereitstellung der Status successful (Erfolgreich) angezeigt wird, können Sie die neue SimplePHPApp-Version folgendermaßen ausführen.

So führen Sie SimplePHPApp aus

1. Klicken Sie auf der Seite Instances (Instances) in der Zeile php-app1 (php-app1) auf die öffentliche IP-Adresse.

   In Ihrem Browser sollte die folgende Seite angezeigt werden.

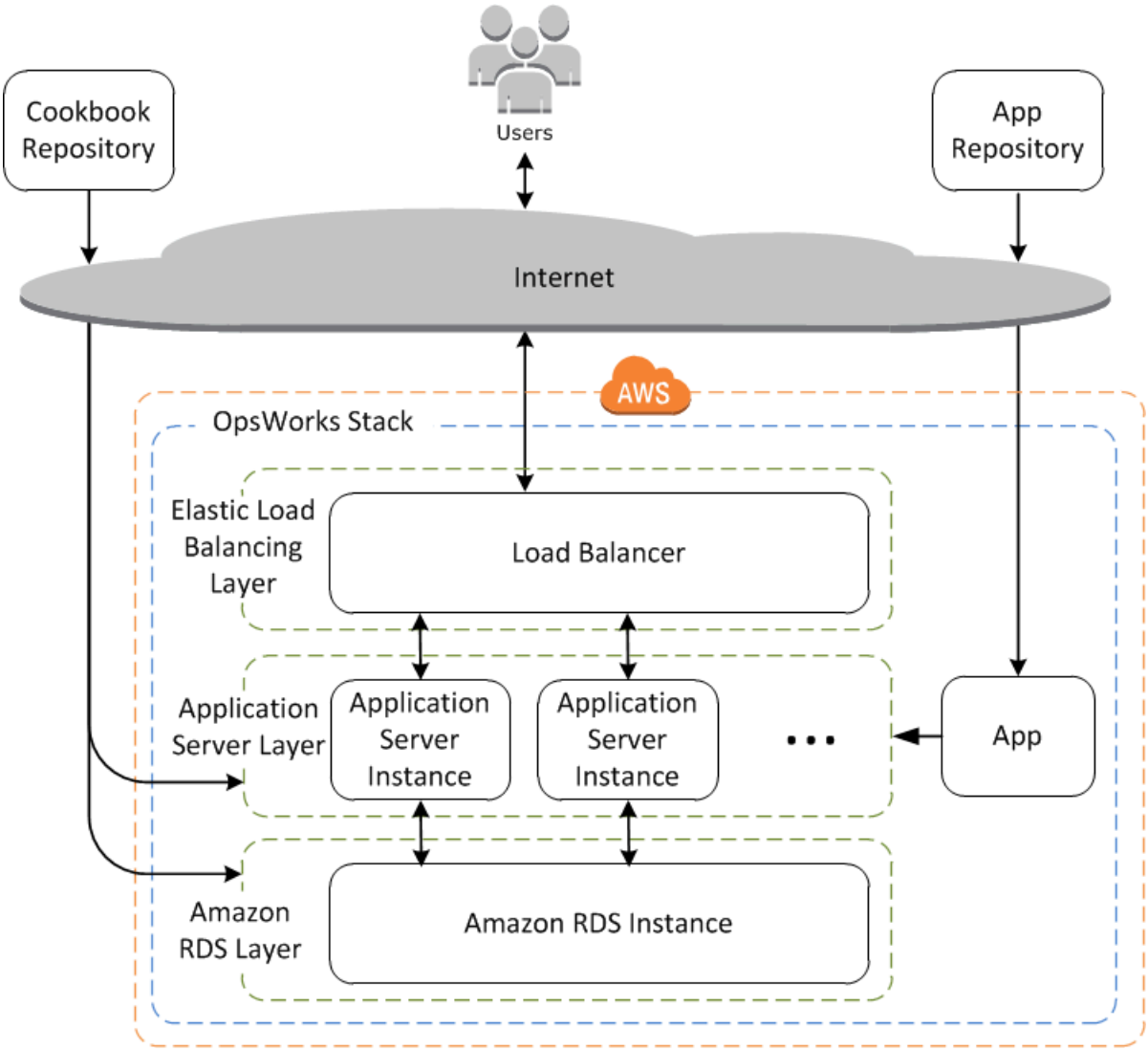
2. Klicken Sie auf Share Your Thought (Kommentar eingeben) und geben Sie z. B. **`Hello world!`** für Your Thought (Ihr Kommentar) und Ihren Namen für Your Name (Ihr Name) ein. Klicken Sie dann auf Submit Your Thought (Kommentar absenden), um die Nachricht zur Datenbank hinzuzufügen.

3. Klicken Sie auf Go Back (Zurück), um alle Nachrichten in der Datenbank anzuzeigen.

Schritt 4: Skalieren MyStack

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

MyStack hat derzeit nur einen Anwendungsserver. Ein Produktions-Stack benötigt wahrscheinlich mehrere Anwendungsserver, um den eingehenden Datenverkehr zu bewältigen, und einen Load Balancer, der diesen Datenverkehr gleichmäßig auf die Anwendungsserver verteilt. Die Architektur wird in etwa wie folgt aussehen:



AWS OpsWorks Stacks macht es einfach, Stapel zu skalieren. In diesem Abschnitt werden die Grundlagen beschrieben, wie Sie einen Stack skalieren können, indem Sie eine zweite rund um die Uhr verfügbare PHP App Server-Instance zu einem Elastic Load Balancer hinzufügen MyStack und beide Instanzen hinter einem Elastic Load Balancing Load Balancer platzieren. Sie können das Verfahren einfach erweitern, um eine beliebige Anzahl von 24/7-Instances hinzuzufügen, oder Sie können zeit- oder lastbasierte Instances verwenden, damit AWS OpsWorks Stacks Ihren Stack automatisch skaliert. Weitere Informationen finden Sie unter Verwaltung der Last mit zeit- und lastbasierten Instanzen.

Schritt 4.1: Hinzufügen eines Load Balancers

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Elastic Load Balancing ist ein AWS-Service, der den eingehenden Anwendungsdatenverkehr automatisch auf mehrere Amazon EC2 EC2-Instances verteilt. Elastic Load Balancing verteilt nicht nur den Traffic, sondern bietet auch folgende Funktionen:

- Erkennt fehlerhafte Amazon EC2 EC2-Instances.

  Er leitet Datenverkehr auf die übrigen fehlerfreien Instances um, bis die Fehler behoben wurden.
- Er skaliert als Reaktion auf den eingehenden Datenverkehr automatisch die Kapazität zur Anforderungsbearbeitung.

> **Note**
>
> Ein Load Balancer hat zwei Anwendungsgebiete. Das offensichtlichere ist eine gleichmäßige Auslastung Ihrer Anwendungsserver zu gewährleisten. Darüber hinaus ziehen viele Websites es vor, die Anwendungsserver und Datenbanken vom direkten Benutzerzugriff zu trennen. Mit AWS OpsWorks Stacks können Sie dies tun, indem Sie Ihren Stack wie folgt in einer Virtual Private Cloud (VPC) mit einem öffentlichen und privaten Subnetz ausführen.
>
> - Dafür müssen sich die Anwendungsserver und Datenbank in einem privaten Subnetz befinden, auf das zwar andere Instances in der VPC, nicht aber Benutzer zugreifen können.
> - Leiten Sie Benutzerdatenverkehr an einen Load Balancer im öffentlichen Subnetz. Von dort aus wird der Datenverkehr an die Anwendungsserver im privaten Subnetz weitergeleitet, die wiederum Antworten an die Benutzer zurückgeben.

Weitere Informationen finden Sie unter Ausführen eines Stacks in einer VPC. Laden Sie die Datei herunter, um eine AWS CloudFormation Vorlage zu erhalten, die das Beispiel in dieser exemplarischen Vorgehensweise auf die `OpsWorksVPCtemplates.zip` Ausführung in einer VPC erweitert.

Elastic Load Balancing wird zwar oft als Ebene bezeichnet, funktioniert aber etwas anders als die anderen integrierten Ebenen. Anstatt eine Ebene zu erstellen und ihr Instances hinzuzufügen, erstellen Sie mithilfe der Amazon EC2 EC2-Konsole einen Elastic Load Balancing Load Balancer und fügen ihn dann einer Ihrer vorhandenen Ebenen hinzu, normalerweise einer Anwendungsserverschicht. AWS OpsWorks Stacks registriert dann die vorhandenen Instances der Ebene beim Service und fügt automatisch alle neuen Instances hinzu. Das folgende Verfahren beschreibt, wie ein Load Balancer zur PHP App MyStack Server-Ebene hinzugefügt wird.

> **Note**
>
> AWS OpsWorks Stacks unterstützt den Application Load Balancer nicht. Sie können Classic Load Balancer nur mit AWS OpsWorks Stacks verwenden.

Um einen Load Balancer an die PHP App Server-Ebene anzuhängen

1. Verwenden Sie die Amazon EC2 EC2-Konsole, um einen neuen Load Balancer für zu erstellen. MyStack Wie Sie dabei genau vorgehen, ist abhängig davon, ob Ihr Konto EC2-Classic unterstützt. Weitere Informationen finden Sie unter Erste Schritte mit Elastic Load Balancing. Wenn Sie den Assistenten Load Balancer erstellen ausführen, konfigurieren Sie den Load Balancer wie folgt:

   Define Load Balancer (Load Balancer definieren)

   Weisen Sie dem Load Balancer einen leicht erkennbaren Namen wie PHP-LB zu, damit er in der Stacks-Konsole leichter auffindbar ist. AWS OpsWorks Klicken Sie anschließend auf Continue (Weiter), um die übrigen Standardeinstellungen zu übernehmen.

   Wenn Sie eine VPC mit mindestens einem Subnetz aus dem Menü Create LB Inside (LB Inside erstellen) auswählen, müssen Sie für jede Availability Zone, an die Datenverkehr über den Load Balancer geleitet werden soll, ein Subnetz auswählen.

Assign Security Groups (Sicherheitsgruppen zuweisen)

Wenn Ihr Konto die Standard-VPC unterstützt, zeigt der Assistent diese Seite an, um die Sicherheitsgruppe des Load Balancers festzulegen. Für EC2 Classic wird diese Seite nicht angezeigt.

Wählen Sie für diese Anleitung default VPC security group (Standard-VPC-Sicherheitsgruppe) aus.

Configure Security Settings (Sicherheitseinstellungen konfigurieren)

Wenn Sie die Option HTTPS für das Load Balancer Protocol (Load-Balancer-Protokoll) auf der Seite Define Load Balancer (Load Balancer definieren) auswählen, müssen Sie auf dieser Seite die Einstellungen für das Zertifikat, die Verschlüsselung und das SSL-Protokoll konfigurieren. Akzeptieren Sie in dieser Anleitung einfach die Standardwerte und wählen Sie Configure Health Check (Zustandsprüfungen konfigurieren) aus.

Konfigurieren von Zustandsprüfungen

Legen Sie den Ping-Path auf **/** fest und übernehmen Sie für die übrigen Einstellungen die Standardwerte.

Add EC2 Instances (EC2-Instances hinzufügen)

Wählen Sie Weiter. AWS OpsWorks Stacks registriert Instances automatisch beim Load Balancer.

Tags hinzufügen

Fügen Sie Tags hinzu, um die Suche zu vereinfachen. Jeder Tag besteht aus einem Schlüssel-Wert-Paar. Sie können für diese Anleitung beispielsweise **`Description`** als Schlüssel und **`Test LB`** als Wert festlegen.

Prüfen

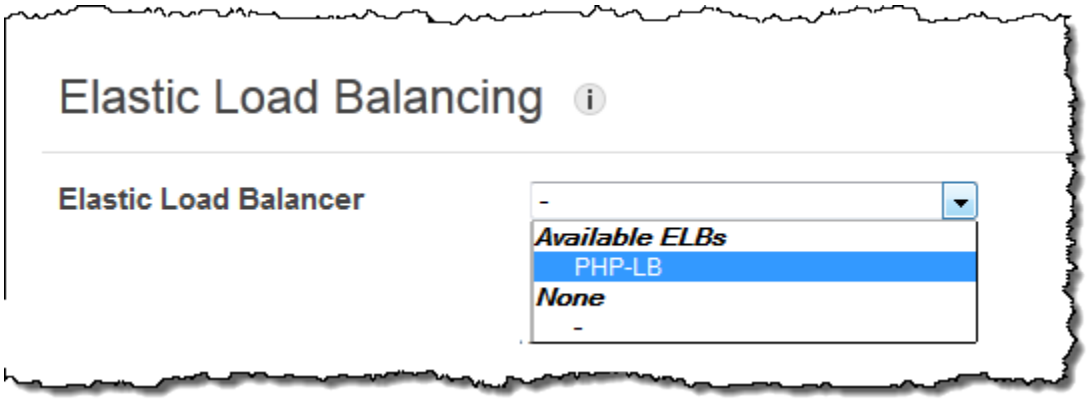
Überprüfen Sie Ihre Auswahl, wählen Sie Create (Erstellen) und dann Close (Schließen) aus, um den Load Balancer zu starten.

2. Wenn Ihr Konto die Standard-VPC unterstützt, müssen Sie nach dem Starten des Load Balancers sicherstellen, dass die Sicherheitsgruppe über die erforderlichen Zugangsregeln verfügt. Mit der Standardregel wird eingehender Datenverkehr nicht akzeptiert.

   1. Wählen Sie im Amazon EC2 EC2-Navigationsbereich die Option Sicherheitsgruppen aus.
   2. Wählen Sie default VPC security group (Standard-VPC-Sicherheitsgruppe) aus.

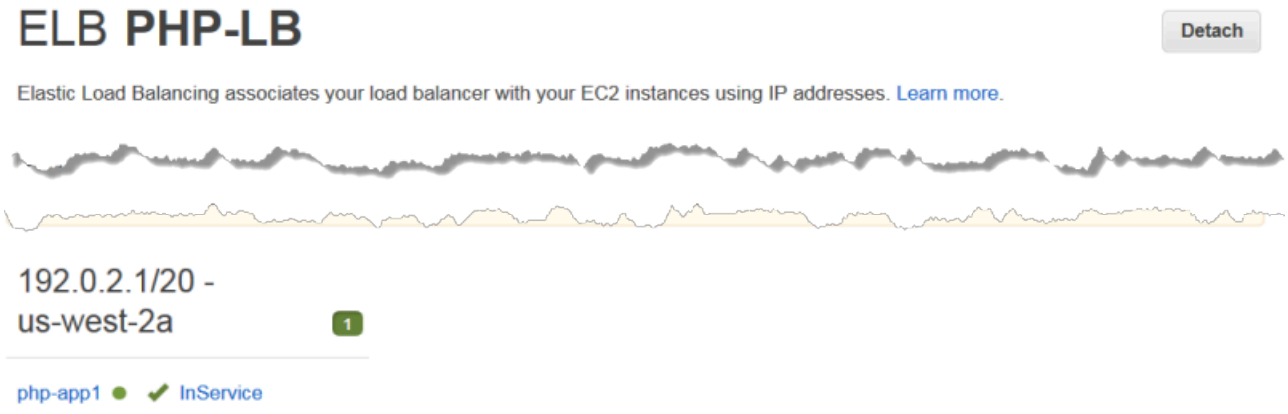
3. Wählen Sie Edit (Bearbeiten) auf der Registerkarte Inbound (Eingehend) aus.
4. Legen Sie für diese Anleitung für Source (Quelle) den Wert Anywhere (Beliebig) fest. So akzeptiert der Load Balancer eingehenden Datenverkehr von beliebigen IP-Adressen.

3. Kehren Sie zur AWS OpsWorks Stacks-Konsole zurück. Wählen Sie auf der Seite Layers (Ebenen) den Link Network (Netzwerk) der Ebene und dann Edit (Bearbeiten) aus.
4. Wählen Sie unter Elastic Load Balancing den Load Balancer aus, den Sie in Schritt 1 erstellt haben, und wählen Sie dann Save (Speichern) aus.



Nachdem Sie den Load Balancer mit dem Layer verbunden haben, registriert AWS OpsWorks Stacks automatisch die aktuellen Instanzen des Layers und fügt neue Instanzen hinzu, sobald sie online sind.

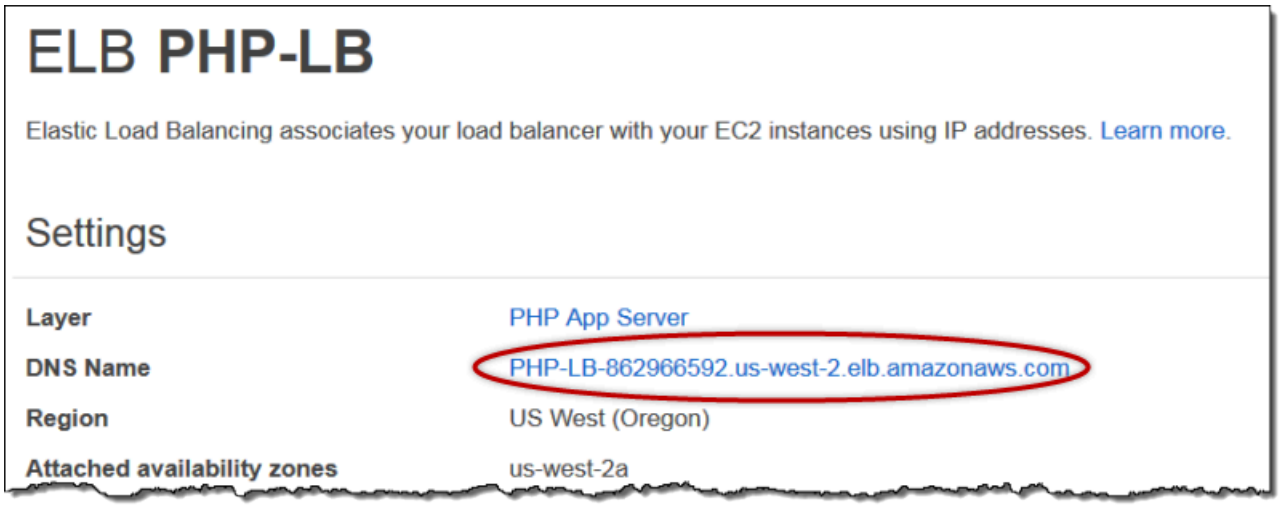
5. Klicken Sie auf der Seite Layers (Ebenen) auf den Load Balancer-Namen, um dessen Detailseite aufzurufen. Wenn die Registrierung abgeschlossen ist und die Instance eine Integritätsprüfung bestanden hat, zeigt AWS OpsWorks Stacks auf der Load Balancer-Seite neben der Instance ein grünes Häkchen an.



Wenn Sie nun eine Anfrage an den Load Balancer senden, können Sie SimplePHPApp ausführen.

So führen Sie SimplePHPApp über den Load Balancer aus

1. Öffnen Sie die Detailseite des Load Balancers erneut, falls sie noch nicht geöffnet ist.
2. Überprüfen Sie auf der Seite "Properties" den Status der Instance und klicken Sie auf den DNS-Namen des Load Balancers, um SimplePHPApp auszuführen. Der Load Balancer leitet die Anfrage an die PHP App Server-Instanz weiter und gibt die Antwort zurück, die genau so aussehen sollte wie die Antwort, die Sie erhalten, wenn Sie auf die öffentliche IP-Adresse der PHP App Server-Instanz klicken.



> **Note**
>
> AWS OpsWorks Stacks unterstützt auch den HAProxy Load Balancer, was für einige Anwendungen Vorteile haben kann. Weitere Informationen finden Sie unter HAProxy Stacks AWS OpsWorks , Ebene.

Schritt 4.2: PHP-App-Server-Instanzen hinzufügen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Jetzt ist der Load Balancer eingerichtet. Sie können den Stack skalieren, indem Sie der PHP App Server-Ebene weitere Instanzen hinzufügen. Aus Ihrer Perspektive schließt dieser Vorgang nahtlos an. Jedes Mal, wenn eine neue PHP App Server-Instanz online geht, registriert AWS OpsWorks Stacks sie automatisch beim Load Balancer und stellt SimplePhpApp bereit, sodass der Server sofort mit der Bearbeitung des eingehenden Datenverkehrs beginnen kann. Der Kürze halber wird in diesem Thema gezeigt, wie eine zusätzliche PHP App Server-Instanz hinzugefügt wird. Sie können jedoch den gleichen Ansatz verwenden, um so viele hinzuzufügen, wie Sie benötigen.

Um dem PHP App Server-Layer eine weitere Instanz hinzuzufügen

1. Klicken Sie auf der Seite Instances unter PHP App Server auf + Instance.
2. Übernehmen Sie die Standardeinstellungen und klicken Sie auf Add Instance (Instance hinzufügen).
3. Klicken Sie auf start (Starten), um die Instance zu starten.

Schritt 4.3: Überwachen MyStack

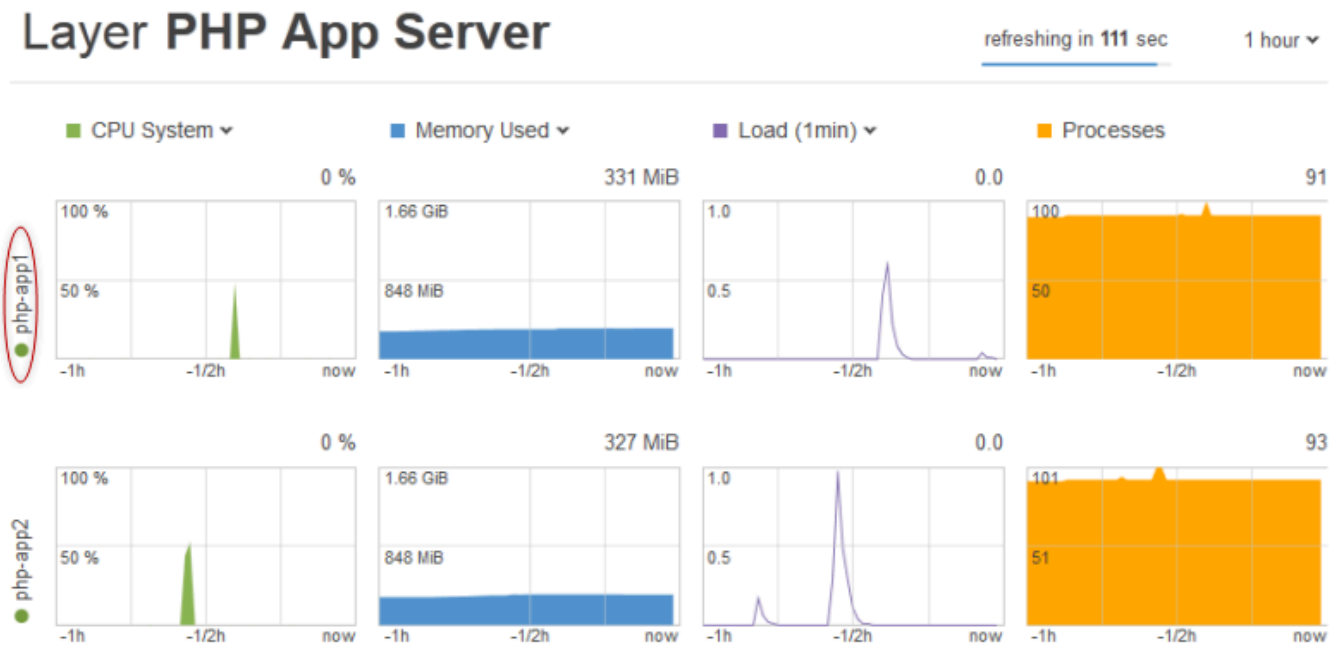
> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks verwendet Amazon CloudWatch , um Metriken für einen Stack bereitzustellen, und fasst sie zur besseren Übersicht auf der Monitoring-Seite zusammen. Hier können Sie Metriken für den gesamten Stack, einen bestimmten Layer oder eine bestimmte Instance betrachten.

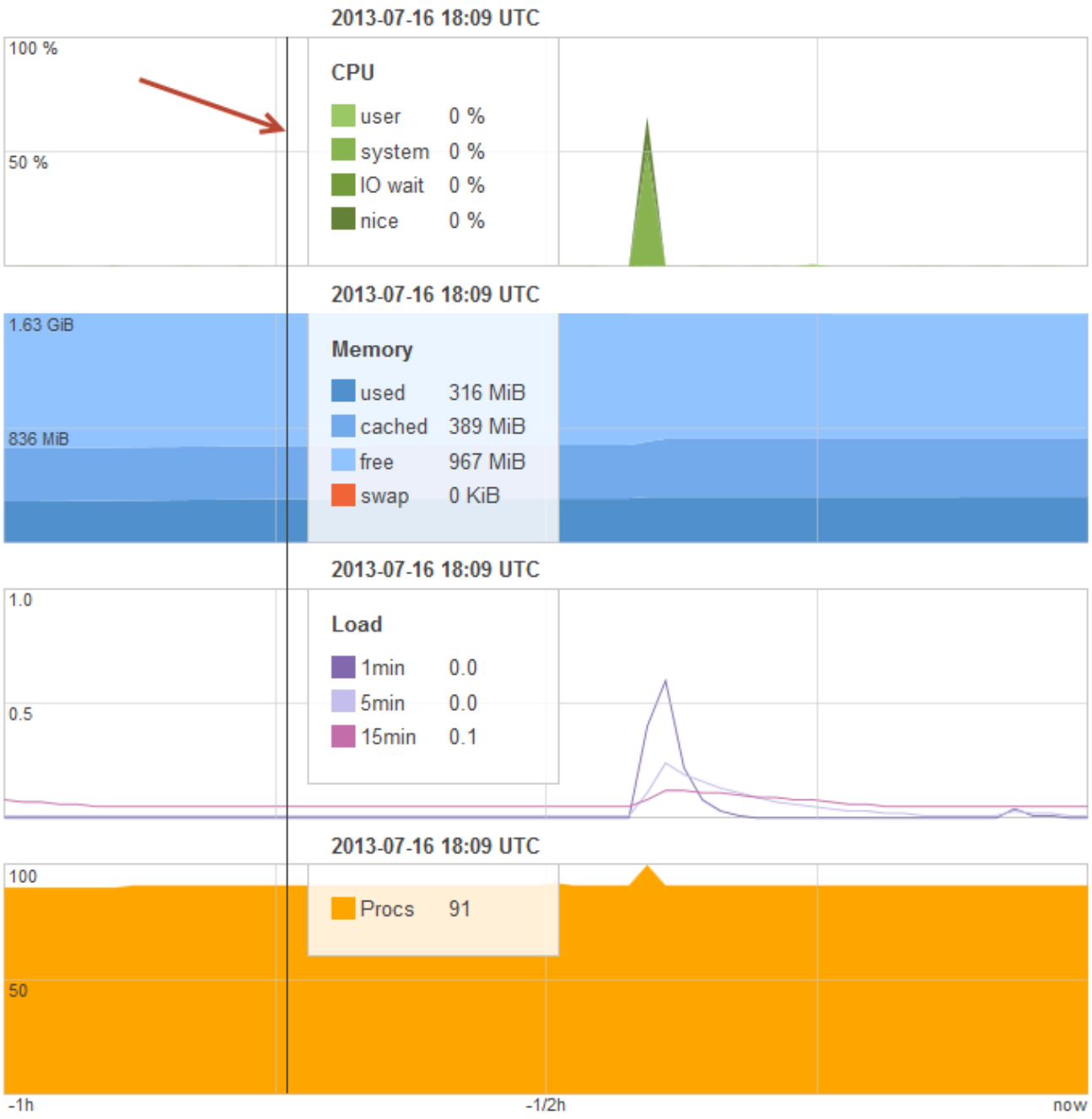
Zur Überwachung MyStack

1. Klicken Sie im Navigationsbereich auf Monitoring (Überwachung), um die durchschnittlichen Metriken der einzelnen Ebenen grafisch darzustellen. Über die Menüs für CPU System (CPU-System), Memory Used (Genutzter Speicher) und Load zeigen Sie weitere zugehörige Metriken an.

2. Klicken Sie auf PHP App Server (PHP-Anwendungsserver), um Metriken für die einzelnen Instances der Ebene anzuzeigen.



3. Klicken Sie auf php-app1 (php-app1), um Metriken für diese Instance anzuzeigen. Bewegen Sie den Schieberegler, um Metriken für einen bestimmten Zeitpunkt anzuzeigen.


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 unterstützt auch den Ganglia-Monitoring-Server, was für einige Anwendungen Vorteile haben kann. Weitere Informationen finden Sie unter Ganglien-Schicht.

Schritt 5: Löschen MyStack

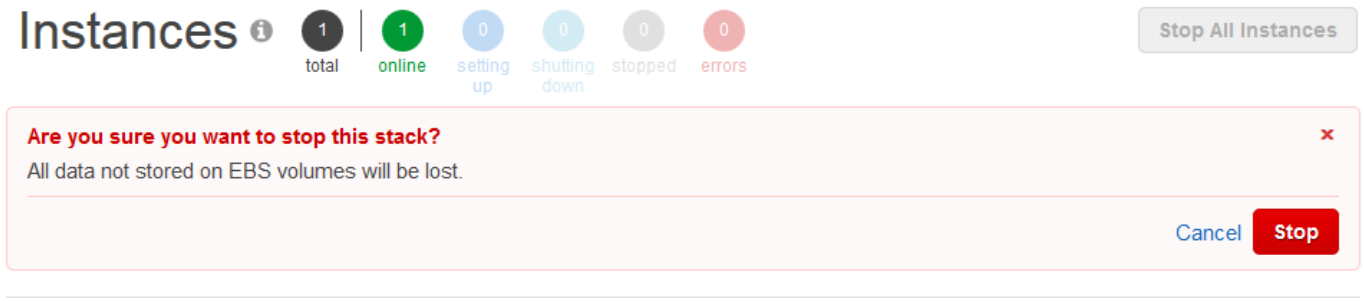
> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sobald Sie AWS-Ressourcen wie Amazon EC2 EC2-Instances nutzen, werden Ihnen Gebühren auf Grundlage Ihrer Nutzung berechnet. Wenn Sie für den Moment fertig sind, sollten Sie die Instances anhalten, sodass keine unerwünschten Gebühren anfallen. Wenn Sie den Stack nicht mehr benötigen, können Sie ihn löschen.

Um zu löschen MyStack

1. Anhalten aller Instances

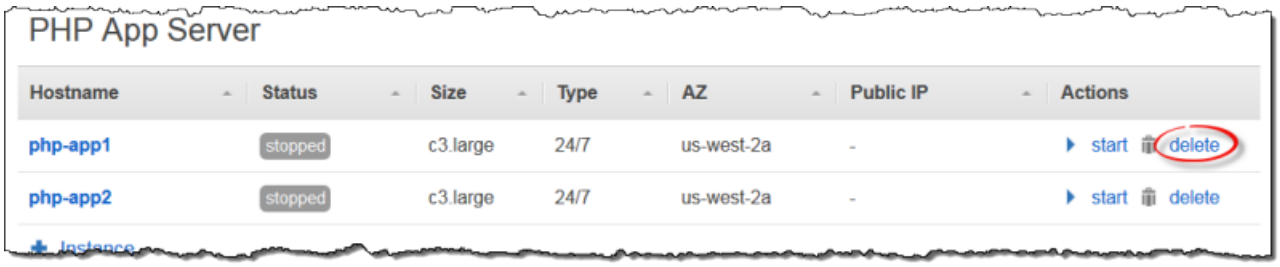
   Klicken Sie auf der Seite Instances auf Stop All Instances (Alle Instances anhalten) und dann auf Stop (Anhalten), wenn Sie aufgefordert werden, den Vorgang zu bestätigen.



   Nachdem Sie auf Stop geklickt haben, beendet AWS OpsWorks Stacks die zugehörigen Amazon EC2 EC2-Instances, jedoch keine zugehörigen Ressourcen wie Elastic IP-Adressen oder Amazon EBS-Volumes.

2. Löschen aller Instances

   Durch das Stoppen der Instance werden lediglich die zugehörigen Amazon EC2 EC2-Instances beendet. Nachdem sich die Instances im Status "Stopped" befinden, müssen Sie alle Instances löschen. Klicken Sie im Layer PHP App Server auf delete in der Spalte Actions der Instance "php-app1".

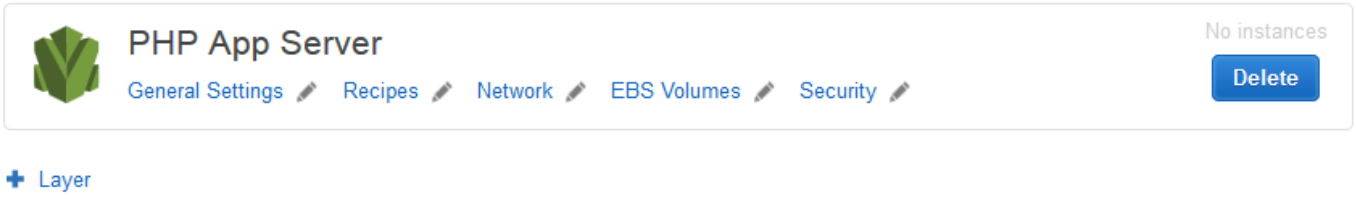


   AWS OpsWorks Stacks fordert Sie dann auf, den Löschvorgang zu bestätigen, und zeigt Ihnen alle abhängigen Ressourcen an. Sie können einzelne oder alle diese Ressourcen behalten. Dieses Beispiel hat keine abhängigen Ressourcen, klicken Sie daher einfach auf Delete (Löschen).

   Wiederholen Sie den Prozess für "php-app2" und die MySQL-Instance, "db-master1". Beachten Sie, dass db-master1 über ein zugeordnetes Amazon Elastic Block Store-Volume verfügt, das standardmäßig ausgewählt ist. Lassen Sie es ausgewählt, um das Volume zusammen mit der Instance zu löschen.

3. Löschen Sie die Layer.

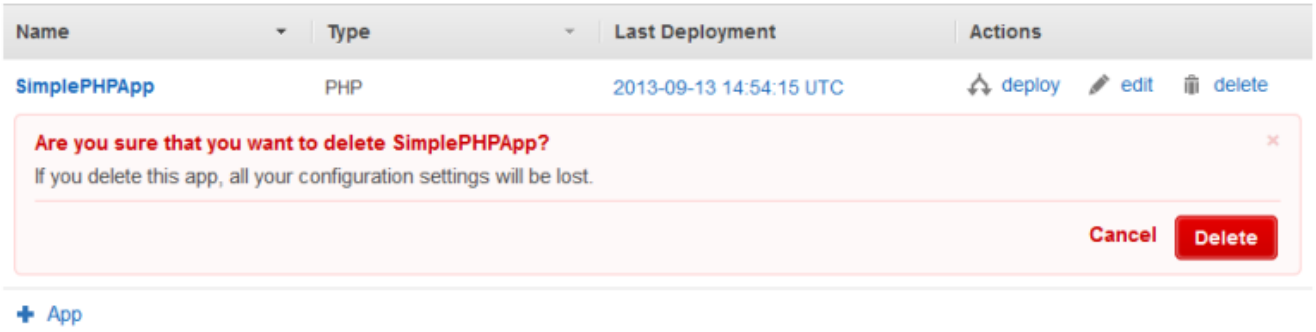
   Klicken Sie auf der Seite Layers (Ebenen) auf Delete (Löschen) und dann zur Bestätigung auf Delete (Löschen).



   Wiederholen Sie den Prozess für die MySQL-Ebene.

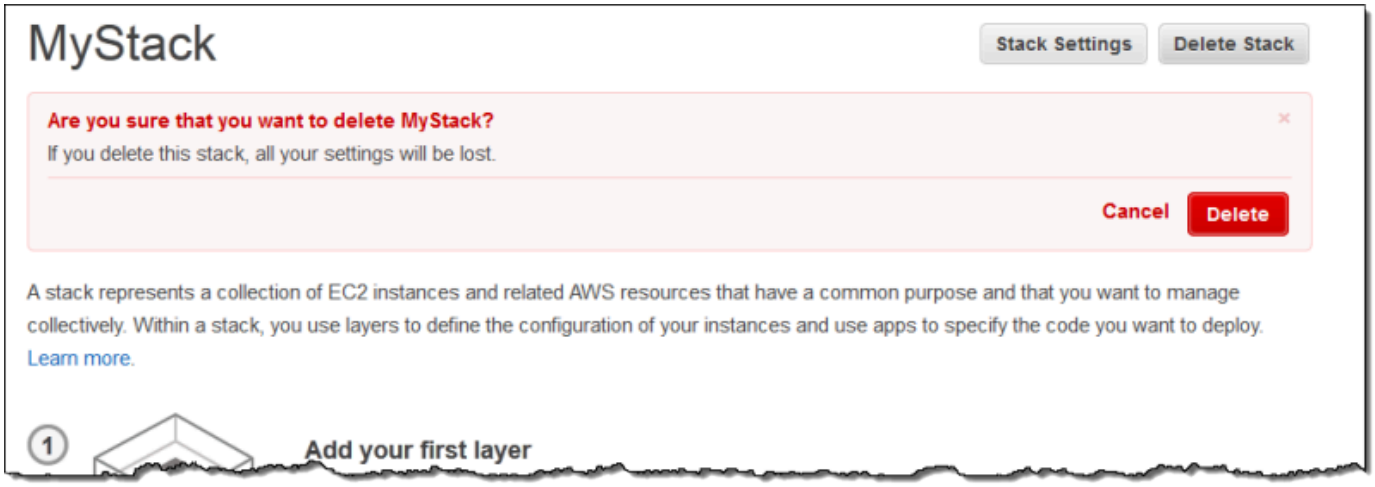
4. Löschen der Anwendung

   Klicken Sie auf der Seite Apps auf delete in der Spalte Actions der App SimplePHPApp und klicken Sie anschließend zur Bestätigung auf Delete.



5. Löschen MyStack

   Klicken Sie auf der Seite Stack auf Delete Stack (Stack löschen) und dann zur Bestätigung auf Delete (Löschen).



Sie haben jetzt das Ende dieser exemplarischen Vorgehensweise erreicht.

Erstellen Ihres ersten Node.js-Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Beispiel wird die Erstellung eines Linux-Stacks, der einen Node.js-Anwendungsserver unterstützt, sowie die Bereitstellung einer einfachen Anwendung beschrieben. Der Stack besteht aus den folgenden Komponenten:

- Eine Node.js App Server-Ebene mit zwei Instanzen
- Ein Elastic Load Balancing Load Balancer zur Verteilung des Datenverkehrs auf die Anwendungsserver-Instances
- Eine Service-Schicht des Amazon Relational Database Service (Amazon RDS), die eine Backend-Datenbank bereitstellt

Themen

Voraussetzungen

In dieser schrittweisen Anleitung wird von Folgendem ausgegangen:

- Sie haben ein AWS-Konto und Grundkenntnisse in der Verwendung von AWS OpsWorks Stacks.

  Wenn Sie neu bei AWS OpsWorks Stacks oder AWS sind, lernen Sie die Grundlagen, indem Sie das Einführungs-Tutorial unter absolvieren. Erste Schritte mit Chef 11 Linux-Stacks
- Sie besitzen grundlegende Kenntnisse über die Implementierung einer Node.js-Anwendung.

  Wenn Sie noch nicht mit Node.js vertraut sind, erlernen Sie die Grundlagen durch Ausführen eines Einführungs-Tutorials, wie zum Beispiel Node: Up and Running.
- Sie haben bereits mindestens einen Stack in der AWS-Region erstellt, die Sie für dieses Beispiel verwenden möchten.

  Wenn Sie den ersten Stack in einer Region erstellen, erstellt AWS OpsWorks Stacks für jeden Layer-Typ eine Amazon Elastic Compute Cloud (Amazon EC2) -Sicherheitsgruppe. Sie benötigen diese Sicherheitsgruppen, um die Amazon RDS-Datenbank-Instance (DB) zu erstellen. Wenn

Sie AWS OpsWorks Stacks noch nicht kennen, empfehlen wir Ihnen, für dieses Beispiel dieselbe Region zu verwenden, die Sie verwendet haben, als Sie das Tutorial unter befolgt haben. Erste Schritte mit Chef 11 Linux-Stacks Wenn Sie eine neue Region verwenden möchten, erstellen Sie einen neuen Stack in der Region. Der Stack muss keine Ebenen oder Instances haben. Sobald Sie den Stack erstellt haben, fügt AWS OpsWorks Stacks der Region automatisch eine Reihe von Sicherheitsgruppen hinzu.

- Sie erstellen Ihren Stack in einer Standard-VPC.

  Sie können EC2-Classic in dieser schrittweisen Anleitung verwenden, jedoch unterscheiden sich einige der Details geringfügig. Mit EC2-Classic geben Sie beispielsweise die Availability Zone (AZ) einer Instance und nicht ihr Subnetz an.

- Ihr IAM-Benutzer hat Vollzugriffsberechtigungen für Stacks. AWS OpsWorks

  Aus Sicherheitsgründen empfehlen wir dringend, dass Sie in dieser schrittweisen Anleitung nicht die Root-Anmeldeinformationen Ihres Kontos verwenden. Erstellen Sie stattdessen einen Benutzer mit Vollzugriffsberechtigungen für AWS OpsWorks Stacks und verwenden Sie diese Anmeldeinformationen mit Stacks. AWS OpsWorks Weitere Informationen finden Sie unter Erstellen eines -Administratorbenutzers.

Implementieren der Anwendung

In dieser exemplarischen Vorgehensweise wird eine einfache Express-Anwendung verwendet, die eine Verbindung zur Amazon RDS-DB-Instance herstellt und die Datenbanken der Instance auflistet.

Zum Implementieren der Anwendung erstellen Sie ein Verzeichnis mit dem Namen `nodedb` an einem geeigneten Speicherort auf Ihrer Workstation und fügen Sie die folgenden drei Dateien hinzu.

Themen

- Paketbeschreibung
- Layoutdatei
- Codedatei

Paketbeschreibung

Fügen Sie die Datei `package.json` mit folgendem Inhalt zum Verzeichnis `nodedb` hinzu, um die Paketbeschreibung der Anwendung zu erstellen. Die Datei `package.json` ist für Express-Anwendungen erforderlich und muss sich im Stammverzeichnis der Anwendung befinden.

```
{
  "name": "Nodejs-DB",
  "description": "Node.js example application",
  "version": "0.0.1",
  "dependencies": {
    "express": "*",
    "ejs": "*",
    "mysql": "*"
  }
}
```

Diese Beispieldatei `package.json` ist relativ minimal gehalten. Sie definiert die erforderlichen Attribute `name` und `version` und gibt die abhängigen Pakete an:

- `express` verweist auf das Express-Paket.
- `ejs` verweist auf das EJS-Paket, das die Anwendung zum Einfügen von Text in eine HTML-Layoutdatei verwendet.
- `mysql` verweist auf das node-mysql-Paket, das die Anwendung zur Verbindungsherstellung mit der RDS-Instance verwendet.

Weitere Informationen zu Paketbeschreibungsdateien finden Sie auf der Website package.json.

Layoutdatei

Um die Layoutdatei der Anwendung zu erstellen, fügen Sie ein `views`-Verzeichnis zum Verzeichnis `nodedb` hinzu. Fügen Sie dann die Datei `views` mit folgendem Inhalt zu `index.html` hinzu:

```
<!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>
```

In diesem Beispiel ist die Layoutdatei ein einfaches HTML-Dokument, das einige Daten von Amazon RDS anzeigt. Jedes `<%= ... =>`-Element stellt jeweils den Wert einer Variablen dar, die in der Codedatei der Anwendung definiert ist. Diese erstellen wir als Nächstes.

Codedatei

Um die Codedatei der Anwendung zu erstellen, fügen Sie eine `server.js`-Datei mit folgendem Inhalt zum Verzeichnis `nodedb` hinzu.

> **Important**
>
> Bei AWS OpsWorks Stacks muss die Hauptcodedatei einer Anwendung Node.js benannt werden `server.js` und sich im Stammordner der Anwendung befinden.

```
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);
```

Im Beispiel werden die Datenbankverbindungsinformationen angegeben, der Datenbankserver abgefragt und die Serverdatenbanken dargestellt. Sie können dieses Beispiel für die Interaktion mit der Datenbank nach Bedarf einfach generalisieren. Die folgenden Hinweise beziehen sich auf die nummerierten Kommentare im vorhergehenden Code.

[1] Einbinden von Datenbankverbindungdaten

Diese `require`-Anweisung enthält die Datenbankverbindungsdaten. Wie später beschrieben, speichert AWS OpsWorks Stacks beim Anhängen einer Datenbankinstanz an eine App die Verbindungsdaten in einer Datei mit dem Name`opsworks.js`, die wie folgt aussieht:

```
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` befindet sich im Verzeichnis `shared/config` der Anwendung, `/srv/www/`*`app_shortname`*`/shared/config`. AWS OpsWorks Stacks fügt jedoch einen symbolischen Link `opsworks.js` in das Stammverzeichnis der Anwendung ein, sodass Sie das Objekt mit nur einbeziehen können. `require 'opsworks'`

[2] Abrufen der Datenbankverbindungsdaten

Mit dieser Gruppe von Anweisungen werden die Verbindungsdaten aus der Datei `opsworks.js` angezeigt, indem die Werte des db-Objekts einer Gruppe von `app.locals`-Eigenschaften zugewiesen werden, die wiederum jeweils einem der Elemente "<%= ... %>" in der Datei `index.html` entsprechen. Das gerenderte Dokument ersetzt die Elemente "<% = ... %>" durch die entsprechenden Eigenschaftswerte.

[3] Herstellen einer Verbindung mit der Amazon RDS-Instance

Im Beispiel wird `node-mysql` für den Zugriff auf die Datenbank verwendet. Zum Herstellen einer Verbindung mit der Datenbank wird im Beispiel ein `connection`-Objekt erstellt, indem die Verbindungsdaten an `createConnection` weitergeleitet werden. Anschließend wird `connection.connect` aufgerufen, um die Verbindung herzustellen.

[4] Abfragen der Datenbank

Nach der Verbindungsherstellung wird im Beispiel `connection.query` für die Abfrage der Datenbank aufgerufen. In diesem Beispiel werden einfach die Datenbanknamen des Servers abgefragt. Die `query`-Methode gibt ein Array von `results`-Objekten (eines für jede Datenbank)

zurück, wobei der Datenbankname der `Database`-Eigenschaft zugewiesen wird. Im Beispiel werden die Namen angefügt und dem Objekt `app.locals.databases,` zugewiesen, mit dem die Liste auf der gerenderten HTML-Seite angezeigt wird.

In diesem Beispiel gibt es fünf Datenbanken, die `nodeexampledb` Datenbank, die Sie bei der Erstellung der RDS-Instance angegeben haben, und vier weitere, die automatisch von Amazon RDS erstellt werden.

[5] Abhören der eingehenden Anforderungen

Die letzte Anweisung hört die eingehenden Anforderungen auf einem bestimmten Port ab. Sie müssen keinen expliziten Portwert festlegen. Wenn Sie die App zu Ihrem Stack hinzufügen, geben Sie an, ob die Anwendung HTTP- oder HTTPS-Anfragen unterstützt. AWS OpsWorks Stacks setzt dann die `PORT` Umgebungsvariable auf 80 (HTTP) oder 443 (HTTPS), und Sie können diese Variable in Ihrer Anwendung verwenden.

Es ist möglich, andere Ports abzuhören, aber die integrierte Sicherheitsgruppe der App Server-Schicht von Node.js, AWS- OpsWorks -NodeJS-App-Server, lässt eingehenden Benutzerverkehr nur zu den Ports 80, 443 und 22 (SSH) zu. Um eingehenden Benutzerverkehr zu anderen Ports zuzulassen, erstellen Sie eine Sicherheitsgruppe mit entsprechenden Regeln für eingehenden Datenverkehr und weisen Sie sie der App Server-Schicht Node.js zu. Ändern Sie keine Regeln für eingehenden Datenverkehr durch Bearbeiten der integrierten Sicherheitsgruppe. Jedes Mal, wenn Sie einen Stack erstellen, überschreibt AWS OpsWorks Stacks die integrierten Sicherheitsgruppen mit den Standardeinstellungen, sodass alle von Ihnen vorgenommenen Änderungen verloren gehen.

> **Note**
>
> Sie können benutzerdefinierte Umgebungsvariablen mit Ihrer Anwendung verknüpfen, wenn Sie die zugehörige Anwendung erstellen oder aktualisieren. Sie können Daten auch mithilfe einer benutzerdefinierten JSON-Datei und eines benutzerdefinierten Rezepts an Ihre Anwendung übertragen. Weitere Informationen finden Sie unter Übermitteln von Daten an Anwendungen.

Erstellen des Datenbankservers und des Load Balancer

In diesem Beispiel werden Amazon RDS-Datenbankserver und Elastic Load Balancing Balancing-Load Balancer-Instances verwendet. Sie müssen jede Instance separat erstellen und dann in Ihren

Stack integrieren. In diesem Abschnitt wird die Erstellung neuer Datenbank- und Load Balancer-Instances erläutert. Sie können aber auch vorhandene Instances verwenden. Wir empfehlen jedoch, dass Sie sich die Verfahren durchlesen, um sicherzustellen, dass diese Instances ordnungsgemäß konfiguriert sind.

Im Folgenden wird die Erstellung einer minimal konfigurierten RDS-DB-Instance beschrieben, die für dieses Beispiel ausreichend ist. Weitere Informationen finden Sie im Amazon RDS-Benutzerhandbuch.

So erstellen Sie die RDS-DB-Instance

1. Öffnen Sie die -Konsole.

   Öffnen Sie die Amazon RDS-Konsole und stellen Sie die Region auf USA West (Oregon) ein. Wählen Sie im Navigationsbereich RDS Dashboard (RDS-Dashboard) und anschließend Launch DB Instance (DB-Instance starten) aus.
2. Legen Sie die Datenbank-Engine fest.

   Wählen Sie MySQL Community Edition (MySQL Community Edition) als Datenbank-Engine aus.
3. Lehnen Sie die Multi-AZ-Bereitstellung ab.

   Wählen Sie No, this instance... (Nein, diese Instance...) und anschließend Next (Weiter) aus. Für dieses Beispiel benötigen Sie keine Multi-AZ-Bereitstellung.
4. Konfigurieren Sie die grundlegenden Einstellungen.

   Legen Sie auf der Seite DB Instance Details (Details für DB-Instance) die folgenden Einstellungen fest:

   - DB Instance Class (DB-Instance-Klasse): db.t2.micro (db.t2.micro)
   - Multi-AZ Deployment (Multi-AZ-Bereitstellung): No (Nein)
   - Allocated Storage (Zugewiesener Speicher): **5** GB
   - DB instance identifier (DB-Instance-Kennung): **`nodeexample`**
   - Master Username (Hauptbenutzername): **`opsworksuser`**.
   - Master Password (Hauptpasswort): Ein Passwort Ihrer Wahl

Notieren Sie die Instance-Kennung, den Benutzernamen und das Passwort zur späteren Verwendung, akzeptieren Sie die Standardeinstellungen für die anderen Optionen und klicken Sie dann auf Next (Weiter).

5. Konfigurieren Sie die erweiterten Einstellungen.

   Legen Sie auf der Seite Configure Advanced Settings (Erweiterte Einstellungen konfigurieren) die folgenden Einstellungen fest:

   - Datenbankname: **`nodeexampledb`**
   - DB-Sicherheitsgruppe (n): AWS- OpsWorks -DB-Master-Server

   > **Note**
   >
   > Die OpsWorksAWS-DB-Master-Server-Sicherheitsgruppe ermöglicht nur den Instances Ihres Stacks den Zugriff auf die Datenbank. Wenn Sie direkt auf die Datenbank zugreifen möchten, fügen Sie eine zusätzliche Sicherheitsgruppe mit den entsprechenden Regeln für eingehenden Datenverkehr an die RDS-DB-Instance an. Weitere Informationen finden Sie unter Amazon RDS-Sicherheitsgruppen. Sie können auch den Zugriff steuern, indem Sie die Instance in einer VPC platzieren. Weitere Informationen finden Sie unter Ausführen eines Stacks in einer VPC.

   Notieren Sie den Datenbanknamen zur späteren Verwendung, akzeptieren Sie die Standardwerte für die anderen Einstellungen und wählen Sie dann Launch DB Instance (DB-Instance starten) aus.

Das folgende Verfahren beschreibt, wie Sie einen Elastic Load Balancing Load Balancer für dieses Beispiel erstellen. Weitere Informationen finden Sie im Elastic Load Balancing-Benutzerhandbuch.

So erstellen Sie den Load Balancer

1. Öffnen Sie die Amazon EC2-Konsole.

   Öffnen Sie die Amazon EC2 EC2-Konsole und stellen Sie sicher, dass die Region auf US West (Oregon) eingestellt ist. Wählen Sie im Navigationsbereich Load Balancers (Load Balancer) und anschließend Create Load Balancer (Load Balancer erstellen) aus.

2. Definieren Sie den Load Balancer.

   Geben Sie auf der Seite Define Load Balancer (Load Balancer definieren) die folgenden Einstellungen an.

   - Name (Name – **`Node-LB`**
   - Create LB Inside — Meine Standard-VPC

   Akzeptieren Sie die Standardeinstellungen für die anderen Optionen und klicken Sie dann auf Next (Weiter).
3. Weisen Sie Sicherheitsgruppen zu.

   Legen Sie auf der Seite Assign Security Groups (Sicherheitsgruppen zuweisen) die folgenden Gruppen fest:

   - default VPC security group (Standard-VPC-Sicherheitsgruppe)
   - OpsWorksAWS-NodeJS-App-Server

   Wählen Sie Weiter aus. Wählen Sie auf der Seite Configure Security Settings (Sicherheitseinstellungen konfigurieren) die Option Next (Weiter) aus. Sie benötigen keinen sicheren Listener für dieses Beispiel.
4. Konfigurieren Sie die Zustandsprüfung.

   Legen Sie auf der Seite Configure Health Check (Zustandsprüfung konfigurieren) die Option Ping Path (Ping-Pfad) auf **`/`** fest und akzeptieren Sie die Standardwerte für die anderen Einstellungen. Wählen Sie Weiter aus. Wählen Sie auf der Seite Add EC2 Instances (EC2-Instances hinzufügen) die Option Next (Weiter) aus. Wählen Sie auf der Seite „Tags hinzufügen“ die Optionen Überprüfen und erstellen aus. AWS OpsWorks Stacks übernimmt die Aufgabe, dem Load Balancer EC2-Instances hinzuzufügen, und für dieses Beispiel benötigen Sie keine Tags.
5. Erstellen Sie den Load Balancer.

   Wählen Sie auf der Seite Review (Prüfen) die Option Create (Erstellen) aus, um den Load Balancer zu erstellen.

Erstellen des Stacks

Jetzt verfügen Sie über alle Komponenten, die zum Erstellen des Stacks erforderlich sind.

So erstellen Sie den Stack

1. Melden Sie sich bei der Stacks-Konsole an AWS OpsWorks .

   Melden Sie sich bei der AWS OpsWorks Stacks-Konsole an und wählen Sie Add Stack (Stack hinzufügen) aus.

2. Erstellen Sie den Stack.

   Um einen neuen Stack zu erstellen, klicken Sie auf Chef 11 stack (Chef 11-Stack) und wählen Sie dann die folgenden Einstellungen aus.

   - – **`NodeStack`**
   - Region — USA West (Oregon)

     Sie können einen Stack in jeder AWS-Region erstellen, wir empfehlen jedoch US West (Oregon) für Tutorials.

   Wählen Sie Add Stack (Stack hinzufügen) aus. Weitere Informationen zu den verschiedenen Stack-Konfigurationseinstellungen finden Sie unter Erstellen eines neuen Stacks.

3. Fügen Sie eine Node.js App Server-Ebene mit einem angeschlossenen Load Balancer hinzu.

   Wählen Sie auf der NodeStackSeite die Option Ebene hinzufügen aus, und geben Sie dann die folgenden Einstellungen an:

   - Layer-Typ — Node.js App Server
   - Elastic Load Balancer — Node-LB

   Akzeptieren Sie die Standardwerte für die anderen Einstellungen und wählen Sie dann Add Layer (Ebene hinzufügen) aus.

4. Fügen Sie Instances zum Layer hinzu und starten Sie sie.

   Wählen Sie im Navigationsbereich die Option Instances (Instances) aus und fügen Sie dann zwei Instances wie folgt zur Rails-App-Serverebene hinzu.

   1. Wählen Sie unter Node.js App Server die Option Add instance aus.

      Legen Sie Size (Größe) auf t2.micro (t2.micro) fest, akzeptieren Sie die Standardwerte für die anderen Einstellungen und wählen Sie dann Add Instance (Instance hinzufügen) aus.

2. Wählen Sie +Instance (+ Instance) aus und fügen Sie eine zweite t2.micro (t2.micro)-Instance in einem anderen Subnetz zur Ebene hinzu.

   Dadurch wird die Instance in einer anderen Availability Zone (AZ) platziert.
3. Wählen Sie Add instance (Instance hinzufügen) aus.
4. Um beide Instances zu starten, wählen Sie Start All Instances (Alle Instances Starten) aus.

Sie haben dieser Ebene einen Elastic Load Balancing Load Balancer zugewiesen. Wenn eine Instance in den Online-Status wechselt oder diesen verlässt, registriert AWS OpsWorks Stacks die Instance automatisch beim Load Balancer oder meldet sie ab.

> **Note**
>
> Für einen Produktions-Stack empfehlen wir, dass Sie Ihre Anwendungsserver-Instances auf mehrere Availability Zones verteilen. Wenn Benutzer keine Verbindung mit einer AZ herstellen können, leitet der Load Balancer den eingehenden Datenverkehr an Instances in den verbleibenden Zonen weiter. Ihre Website funktioniert weiterhin.

5. Registrieren Sie die RDS-DB-Instance beim Stack.

   Wählen Sie im Navigationsbereich die Option Resources (Ressourcen) aus und registrieren Sie die RDS-DB-Instance wie folgt beim Stack.

   1. Wählen Sie die Registerkarte RDS (RDS) und anschließend Show Unregistered RDS DB instances (Nicht registrierte RDS-DB-Instances anzeigen) aus.
   2. Wählen Sie die Instance nodeexampledb (nodeexampledb) aus und legen Sie dann die folgenden Einstellungen fest:
      - Benutzer — Der Master-Benutzername, den Sie bei der Erstellung der Instanz angegeben haben; in diesem Beispiel. **`opsworksuser`**.
      - Passwort — Das Master-Passwort, das Sie bei der Erstellung der Instanz angegeben haben.
   3. Wählen Sie Bei Stack registrieren, um die RDS-DB-Instance dem Stack als Amazon RDS-Service-Layer hinzuzufügen.

> **Warning**
>
> AWS OpsWorks Stacks validiert die Benutzer - oder Passwortwerte nicht, sondern leitet sie einfach an die Anwendung weiter. Wenn Sie sie falsch eingeben, kann Ihre Anwendung keine Verbindung mit der Datenbank herstellen.

Um die RDS-DB-Instance als Amazon RDS-Service-Layer zum Stack hinzuzufügen, wählen Sie Mit Stack registrieren.

Bereitstellen der Anwendung

Sie müssen die Anwendung in einem Remote-Repository speichern. Bei der Bereitstellung stellt AWS OpsWorks Stacks den Code und die zugehörigen Dateien aus dem Repository auf den Anwendungsserver-Instances bereit. Der Einfachheit halber verwendet dieses Beispiel ein öffentliches Amazon Simple Storage Service (Amazon S3) -Archiv als Repository. Sie können aber auch mehrere andere Repository-Typen verwenden, darunter Git und Subversion. Weitere Informationen finden Sie unter Anwendungsquelle.

So stellen Sie die Anwendung bereit

1. Bündeln Sie die Anwendung in eine Archivdatei.

   Erstellen Sie ein `.zip`-Archiv des Verzeichnisses `nodedb` und zugehöriger Unterverzeichnisse und nennen Sie es "nodedb.zip". Sie können auch andere Archivierungsdateitypen verwenden, einschließlich gzip, bzip2 und Tarball. Beachten Sie, dass AWS OpsWorks Stacks keine unkomprimierten Tarballs unterstützt. Weitere Informationen finden Sie unter Anwendungsquelle.
2. Laden Sie die Archivdatei auf Amazon S3 hoch.

   Laden `nodedb.zip` Sie in einen Amazon S3 S3-Bucket hoch, machen Sie die Datei öffentlich und kopieren Sie die URL der Datei zur späteren Verwendung. Weitere Informationen zum Erstellen von Buckets und Hochladen von Dateien finden Sie unter Erste Schritte mit Amazon Simple Storage Service

> **Note**
>
> AWS OpsWorks Stacks können auch private Dateien aus einem Amazon S3 S3-Bucket bereitstellen, aber der Einfachheit halber verwendet dieses Beispiel eine öffentliche Datei. Weitere Informationen finden Sie unter Anwendungsquelle.

3. Erstellen Sie eine AWS OpsWorks Stacks-App.

   Kehren Sie zur AWS OpsWorks Stacks-Konsole zurück, wählen Sie im Navigationsbereich Apps und dann App hinzufügen aus. Nehmen Sie folgende Einstellungen vor:

   - Name (Name – `NodeDB`.

     Bei dieser Zeichenfolge handelt es sich um den Anzeigenamen der Anwendung. Für die meisten Zwecke benötigen Sie den Kurznamen der App, den AWS OpsWorks Stacks aus dem Anzeigenamen generiert, indem alle Zeichen in Kleinbuchstaben umgewandelt und Satzzeichen entfernt werden. In diesem Beispiel lautet die Kurzbezeichnung `nodedb`. Um die Kurzbezeichnung einer Anwendung zu überprüfen, wählen Sie die Anwendung nach dem Erstellen auf der Seite Apps (Anwendungen) aus, um ihre Detailseite anzuzeigen.
   - Typ – `Node.js`.
   - Datenquellentyp – `RDS`.
   - Datenbank-Instance — Wählen Sie die Amazon RDS-DB-Instance aus, die Sie zuvor registriert haben.
   - Database name: Geben Sie den für dieses Beispiel zuvor erstellten Datenbanknamen `nodeexampledb` an.
   - Repository-Typ – `Http Archive`.

     Sie müssen diesen Repository-Typ für öffentliche Amazon S3 S3-Dateien verwenden. Der Typ `S3 Archive` wird nur für private Archive verwendet.
   - Repository-URL — Die Amazon S3-URL der Archivdatei.

   Verwenden Sie die Standardwerte für die übrigen Einstellungen und klicken Sie anschließend auf Add App (App hinzufügen), um die Anwendung zu erstellen.

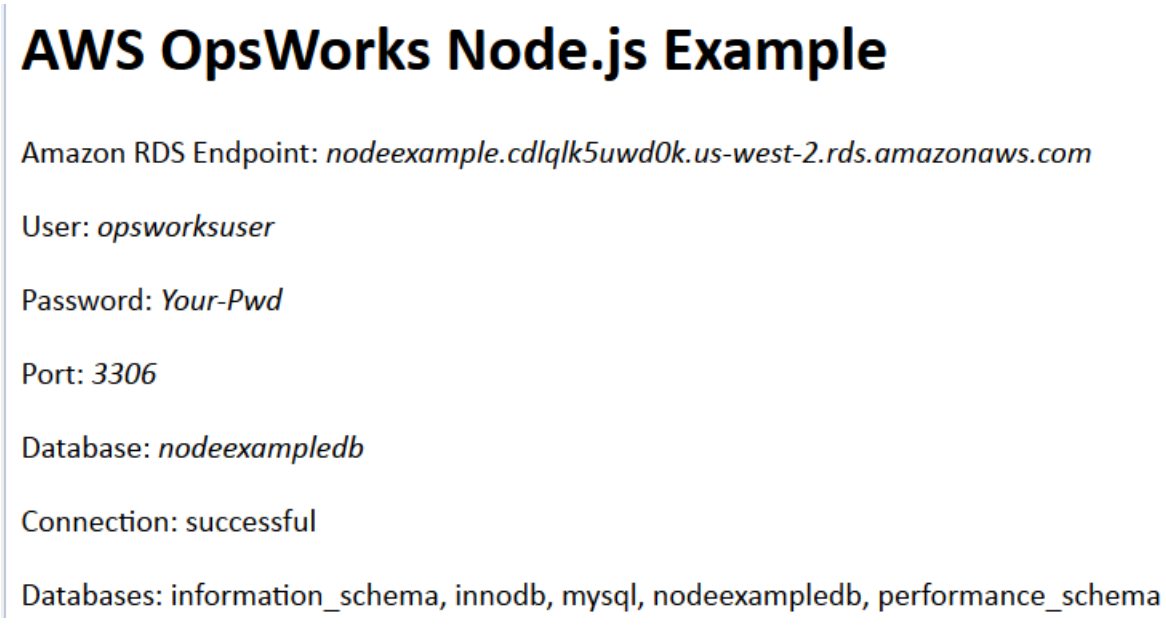
4. Stellen Sie die Anwendung bereit.

   Öffnen Sie die Seite Apps (Anwendungen) und wählen Sie in der Spalte Actions (Aktionen) der NodeDB-Anwendung die Option deploy (Bereitstellen) aus. Wählen Sie dann Deploy, um die App auf den Server-Instances bereitzustellen. AWS OpsWorks Stacks führt die Deploy-Rezepte auf jeder Instanz aus, wodurch die Anwendung aus dem Repository heruntergeladen und der Server neu gestartet wird. Wenn jede Instance ein grünes Häkchen aufweist und der Status als successful (Erfolgreich) angegeben ist, ist die Bereitstellung abgeschlossen und die Anwendung kann nun Anforderungen verarbeiten.

   > **Note**
   >
   > Wenn die Bereitstellung fehlschlägt, wählen Sie in der Spalte Log (Protokoll) die Option show (Anzeigen) aus, um das Chef-Protokoll der Bereitstellung anzuzeigen. Die Fehlerinformationen werden unten angegeben.

5. Öffnen Sie die Anwendung .

   Wählen Sie zum Öffnen der Anwendung Layers (Ebenen) aus. Wählen Sie den Load Balancer und dann dessen DNS-Namen aus, der eine HTTP-Anforderung an den Load Balancer sendet. Dies sollte etwa wie folgt aussehen.



> **Note**
>
> AWS OpsWorks Stacks stellt Apps während der Einrichtung automatisch auf neuen Instanzen bereit. Die manuelle Bereitstellung ist nur für Online-Instances erforderlich. Weitere Informationen finden Sie unter Bereitstellen von Anwendungen. Eine allgemeine Beschreibung der Bereitstellung, einschließlich einiger komplexer Bereitstellungsstrategien, finden Sie unter Verwalten und Bereitstellen von Anwendungen und Rezeptbüchern.

Wie geht es weiter?

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In dieser schrittweisen Anleitungen wurden Sie durch die Grundlagen der Einrichtung eines einfachen Node.js-Anwendungsserver-Stacks geführt. Im Folgenden finden Sie einige Vorschläge für die nächsten Schritte.

Überprüfen der integrierten Node.js-Rezeptbücher

Wenn Sie wissen möchten, wie die Instanzen im Detail konfiguriert sind, schauen Sie sich das integrierte Kochbuch der Ebene, opsworks_nodejs, an, das die Rezepte und zugehörigen Dateien enthält, die AWS OpsWorks Stacks zur Installation und Konfiguration der Software verwendet, sowie das integrierte Deploy-Cookbook, das die Rezepte enthält, die Stacks zur Bereitstellung der Apps verwendet. AWS OpsWorks

Anpassen der Serverkonfiguration

Der Beispiel-Stack ist recht einfach. Für die Produktionsnutzung sollten Sie den Stack anpassen. Weitere Informationen finden Sie unter Stacks anpassen AWS OpsWorks.

Hinzufügen der SSL-Unterstützung

Sie können die SSL-Unterstützung für Ihre App aktivieren und AWS OpsWorks Stacks bei der Erstellung der App die entsprechenden Zertifikate zur Verfügung stellen. AWS OpsWorks Stacks

installiert dann die Zertifikate im entsprechenden Verzeichnis. Weitere Informationen finden Sie unter Verwenden von SSL.

Hinzufügen von In-Memory-Daten-Caching

Produktions-Websites verbessern häufig die Leistung durch Zwischenspeicherung von Daten in einem Hauptspeicher-basierten Key-Value Store, z. B. Redis oder Memcache. Sie können beide mit einem AWS OpsWorks Stacks-Stack verwenden. Weitere Informationen finden Sie unter ElastiCache Redis und Memcached.

Verwenden einer komplexeren Bereitstellungsstrategie

Im Beispiel wird eine einfache Anwendungsbereitstellungsstrategie verwendet, mit der die Aktualisierung für alle Instances gleichzeitig bereitgestellt wird. Diese Methode ist einfach und schnell, es darf jedoch kein Fehler unterlaufen. Wenn die Bereitstellung fehlschlägt oder bei der Aktualisierung Probleme auftreten, könnte sich dies auf alle Instances in Ihrem Produktions-Stack auswirken. Möglicherweise wird Ihre Website unterbrochen oder deaktiviert, bis Sie das Problem beheben können. Weitere Informationen zu Bereitstellungsstrategien finden Sie unter Verwalten und Bereitstellen von Anwendungen und Rezeptbüchern.

Erweitern Sie die App-Server-Ebene von Node.js

Sie können die Ebene auf unterschiedliche Weise erweitern. Sie können beispielsweise Rezepte zum Ausführen von Skripts auf den Instances oder Chef-Bereitstellungs-Hooks zum Anpassen der Anwendungsbereitstellung implementieren. Weitere Informationen finden Sie unter Erweitern eines Layers.

Definieren der Umgebungsvariablen

Sie können Daten an Ihre Anwendung übertragen, indem Sie die Umgebungsvariablen für die zugehörige Anwendung definieren. Wenn Sie die App bereitstellen, exportiert AWS OpsWorks Stacks diese Variablen, sodass Sie von Ihrer App aus darauf zugreifen können. Weitere Informationen finden Sie unter Verwenden von -Umgebungsvariablen.

Stacks anpassen AWS OpsWorks

Important

Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Die integrierten Ebenen von Stacks bieten Standardfunktionen, die für viele Zwecke ausreichend sind. Möglicherweise werden Sie jedoch auf folgende Probleme stoßen:

- Die Standardkonfiguration eines integrierten Layers ist zwar ausreichend, aber nicht ideal, und Sie möchten sie an Ihre speziellen Anforderungen anpassen.

  Möglicherweise möchten Sie beispielsweise die Nginx-Serverkonfiguration eines statischen Webserver-Layers optimieren, indem Sie Ihre eigenen Werte für Einstellungen wie die maximale Anzahl von Worker-Prozessen oder den Wert angeben. `keepalivetimeout`
- Die Funktionalität eines integrierten Layers ist in Ordnung, Sie möchten jedoch weitere Pakete installieren oder benutzerdefinierte Installationsskripte ausführen, um die Funktionalität zu erweitern.

  Beispielsweise möchten Sie möglicherweise eine PHP-App-Server-Ebene erweitern, indem Sie auch einen Redis-Server installieren.
- Ihre Anforderungen werden über die integrierten Layer nicht erfüllt.

  AWS OpsWorks Stacks enthält beispielsweise keine integrierten Ebenen für einige beliebte Datenbankserver. Sie können daher einen benutzerdefinierten Layer erstellen, der diese Server auf den Instances des Layers installiert.
- Sie führen einen Windows-Stack aus, der nur benutzerdefinierte Layer unterstützt.

AWS OpsWorks Stacks bietet eine Vielzahl von Möglichkeiten, Ebenen an Ihre spezifischen Anforderungen anzupassen. Die nachfolgenden Beispiele werden zunehmend komplexer und leistungsfähiger:

> **Note**
>
> Einige dieser Ansätze können nur auf Linux-Stacks ausgeführt werden. Weitere Informationen finden Sie in den folgenden Themen.

- Verwenden Sie benutzerdefiniertes JSON, um die Standardeinstellungen von AWS OpsWorks Stacks zu überschreiben.

- Implementieren Sie ein benutzerdefiniertes Chef-Kochbuch mit einer Attributdatei, die die Standardeinstellungen AWS OpsWorks von Stacks überschreibt.
- Implementieren Sie ein benutzerdefiniertes Chef-Kochbuch mit einer Vorlage, die eine Standard-Stacks-Vorlage überschreibt oder erweitert. AWS OpsWorks
- Implementieren Sie ein benutzerdefiniertes Chef-Rezeptbuch mit einem einfachen Rezept, um ein Shell-Skript auszuführen.
- Implementieren Sie ein benutzerdefiniertes Chef-Rezeptbuch mit Rezepten, die Aufgaben wie das Erstellen und Konfigurieren von Verzeichnissen, Installieren von Paketen, Erstellen von Konfigurationsdateien, Bereitstellen von Apps usw. übernehmen.

Sie können abhängig von der Chef-Version und dem Betriebssystem des Stacks Rezepte auch überschreiben.

- Bei Chef 0.9- und Chef 11.4-Stacks ist es nicht möglich, integrierte Rezepte zu überschreiben, indem Sie ein benutzerdefiniertes Rezept mit demselben Rezeptbuch- und Rezeptnamen implementieren.

  Für jedes Lebenszyklusereignis führt AWS OpsWorks Stacks immer zuerst die integrierten Rezepte aus, gefolgt von allen benutzerdefinierten Rezepten. Da in diesem Chef-Versionen Rezepte mit demselben Rezeptbuch- und Rezeptnamen nicht mehrfach ausgeführt werden, hat das integrierte Rezept Priorität und das benutzerdefinierte Rezept wird nicht ausgeführt.
- Auf Chef 11.10-Stacks können Sie integrierte Rezepte überschreiben.

  Weitere Informationen finden Sie unter Installation und Vorrang von Rezeptbüchern.
- Auf Windows-Stacks können Sie integrierte Rezepte nicht überschreiben.

  Die Art und Weise, wie AWS OpsWorks Stacks Chef-Läufe für Windows-Stacks behandelt, erlaubt nicht, dass integrierte Rezepte außer Kraft gesetzt werden.

> **Note**
>
> Da viele der Techniken benutzerdefinierte Kochbücher verwenden, sollten Sie zuerst lesen, Cookbooks und Rezepte wenn Sie mit der Implementierung von Kochbüchern noch nicht vertraut sind. Rezeptbücher – Grundlagenbietet eine ausführliche Einführung in die Implementierung benutzerdefinierter Kochbücher und Implementierung von Kochbüchern für

Stacks AWS OpsWorks behandelt einige Details zur Implementierung von Kochbüchern für Stacks-Instanzen. AWS OpsWorks

Themen

Anpassen der AWS OpsWorks Stacks-Konfiguration durch Überschreiben von Attributen

Important

Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Für Windows-Stacks und Chef 12-Linux-Stacks verwendet Stacks separate Chef-Läufe für integrierte Rezepte und benutzerdefinierte Rezepte. AWS OpsWorks Dies bedeutet, dass Sie die in diesem Abschnitt erläuterten Techniken nicht verwenden können, um integrierte Attribute für Windows- und Chef 12 Linux-Stacks zu überschreiben.

Rezepte und Vorlagen sind abhängig von einer Vielzahl von Chef-Attributen für Instance- oder Stack-spezifische Informationen wie Layer-Konfigurationen oder Servereinstellungen der Anwendung. Diese Attribute haben mehrere Quellen:

- Benutzerdefiniertes JSON — Sie können optional benutzerdefinierte JSON-Attribute angeben, wenn Sie einen Stack erstellen, aktualisieren oder klonen oder wenn Sie eine App bereitstellen.

- Stack-Konfigurationsattribute —AWS OpsWorks Stacks definiert diese Attribute so, dass sie Informationen zur Stack-Konfiguration enthalten, einschließlich der Informationen, die Sie in den Konsoleneinstellungen angeben.
- Bereitstellungsattribute — AWS OpsWorks definiert bereitstellungsbezogene Attribute für Deploy-Ereignisse.
- Kochbuchattribute — Integrierte und benutzerdefinierte Kochbücher enthalten in der Regel eine oder mehrere Attributdateien, die Attribute enthalten, die kochbuchspezifische Werte darstellen, z. B. Konfigurationseinstellungen für Anwendungsserver.
- Chef — Das Ohai-Tool von Chef definiert Attribute, die eine Vielzahl von Systemkonfigurationseinstellungen repräsentieren, wie z. B. den CPU-Typ und den installierten Speicher.

Eine vollständige Liste der Attribute für Stack-Konfigurationen, Bereitstellungen und integrierte Rezeptbücher finden Sie unter Stack-Konfigurations- und Bereitstellungsattribute: Linux und Integrierte Rezeptbuchattribute. Weitere Informationen zu Ohai-Attributen finden Sie unter Ohai.

Wenn ein Lebenszyklusereignis wie z. B. Bereitstellen oder Konfigurieren auftritt oder wenn Sie einen Stack-Befehl wie z. B. `execute_recipes` oder `update_packages` ausführen, reagiert AWS OpsWorks Stacks folgendermaßen:

- Sendet einen entsprechenden Befehl an den Agent jeder betroffenen Instance.

  Der Agent führt die entsprechenden Rezepte aus. Für ein Bereitstellungsereignis beispielsweise führt der Agent die integrierten Bereitstellungsrezepte aus, gefolgt von den benutzerdefinierten Bereitstellungsrezepten.
- Führt benutzerdefinierte JSON- und Bereitstellungsattribute mit den Stack-Konfigurationsattributen aus und installiert sie auf den Instances.

Die Attribute des benutzerdefinierten JSON-Objekts, der Stack-Konfiguration, der Bereitstellung und Rezeptbuchattribute und Ohai-Attribute werden in ein Knotenobjekt eingeführt, das den Rezepten Attributwerte zuordnet. Eine Instance ist im Wesentlichen zustandslos, was die Stack-Konfigurationsattribute betrifft, einschließlich aller benutzerdefinierten JSON-Objekte. Wenn Sie einen Bereitstellungs- oder Stack-Befehl ausführen, verwenden die zugehörigen Rezepte die Attribute der Stack-Konfiguration, die mit dem Befehl heruntergeladen wurden.

Themen

- Priorität von Attributen
- Überschreiben von Attributen mit einem benutzerdefinierten JSON-Objekt
- Überschreiben von AWS OpsWorks Stacks-Attributen mithilfe von benutzerdefinierten Cookbook-Attributen

Priorität von Attributen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn ein Attribut eindeutig definiert ist, wird es von Chef einfach in das Knotenobjekt integriert. Allerdings kann jede Attributquelle jedes Attribut definieren, sodass das gleiche Attribut mehrere Definitionen mit unterschiedlichen Werten haben kann. Das integrierte `apache2`-Rezeptbuch definiert beispielsweise `node[:apache][:keepalive]`, aber Sie können das Attribut auch in einem benutzerdefinierten JSON-Objekt oder in einem benutzerdefinierten Rezeptbuch definieren. Wenn ein Attribut mehrere Definitionen hat, werden sie in einer Reihenfolge beschrieben, die später festgelegt wird. Das Knotenobjekt erhält die Definition mit der höchsten Priorität.

Ein Attribut wird wie folgt definiert:

```
node.type[:attribute][:sub_attribute][:...]=value
```

Wenn ein Attribut mehrere Definitionen hat, bestimmt der Typ, welche Definition Vorrang hat, und diese Definition wird in das Knotenobjekt aufgenommen. AWS OpsWorks Stacks verwendet die folgenden Attributtypen:

- default — Dies ist der gebräuchlichste Typ und bedeutet im Wesentlichen „diesen Wert verwenden, wenn das Attribut noch nicht definiert wurde“. Wenn alle Definitionen eines Attributs den `default`-Typ haben, hat die erste Definition in der Auswertungsreihenfolge Vorrang und nachfolgende Werte werden ignoriert. Beachten Sie, dass AWS OpsWorks Stacks alle Definitionen der Stack-Konfiguration und der Bereitstellungsattribute auf `default` Typ festlegt.

- normal — Attribute dieses Typs haben Vorrang vor allen `default` oder `normal` Attributen, die zu einem früheren Zeitpunkt in der Bewertungsreihenfolge definiert wurden. Wenn z. B. das erste Attribut aus einem integrierten Rezeptbuch stammt und einen `default`-Typ hat und das zweite ein vom Benutzer definiertes Attribut vom `normal`-Typ ist, hat die zweite Definition Vorrang.
- set — Dies ist ein veralteter Typ, den Sie möglicherweise in älteren Kochbüchern finden. Es wurde durch den `normal`-Typ ersetzt, der dieselbe Priorität hat.

Chef unterstützt mehrere zusätzliche Attributtypen, einschließlich einem `automatic`-Typ, der Vorrang vor allen anderen Attributdefinitionen hat. Die vom Ohai Tool generierten Attributdefinitionen sind alle `automatic`-Typen, also tatsächlich schreibgeschützt. Dies ist normalerweise kein Problem, da es keinen Grund gibt, sie zu überschreiben, und sie sich von den Attributen von Stacks unterscheiden. AWS OpsWorks Sie sollten jedoch darauf achten, Ihre benutzerdefinierten Rezeptbuchattribute zu benennen, sodass sie sich von den Ohai-Attributen unterscheiden. Weitere Informationen zu Attributen finden Sie unter About Attributes.

> **Note**
>
> Das Ohai Tool ist eine ausführbare Datei, die Sie über die Befehlszeile ausführen können. Um die Ohai-Attribute einer Instance aufzulisten, melden Sie sich bei der Instance an und führen Sie `ohai` in einem Terminalfenster aus. Beachten Sie, dass eine sehr lange Ausgabe produziert wird.

Hier sehen Sie die Schritte, durch die die verschiedenen Attributdefinitionen in das Knotenobjekt integriert werden:

1. Führen Sie alle Attribute der benutzerdefinierten Stack-Konfiguration mit den Attributen der Stack-Konfiguration und der Bereitstellung zusammen.

   Benutzerdefinierte JSON-Attribute können für den Stack oder für eine bestimmte Bereitstellung festgelegt werden. Sie stehen an erster Stelle der Auswertungsreihenfolge und sind tatsächlich `normal`-Typen. Wenn ein oder mehrere Stack-Konfigurationsattribute auch in einem benutzerdefinierten JSON-Objekt definiert sind, werden die benutzerdefinierten JSON-Werte vorgezogen. Andernfalls integriert AWS OpsWorks Stacks die benutzerdefinierten JSON-Attribute einfach in die Stack-Konfiguration.
2. Führen Sie alle benutzerdefinierten JSON-Bereitstellungsattribute mit den Attributen der Stack-Konfiguration und der Bereitstellung zusammen.

Benutzerdefinierte JSON-Bereitstellungsattribute sind außerdem tatsächlich `normal`-Typen, sodass sie vor integrierten und benutzerdefinierten JSON-Stack-Konfigurationen und integrierten JSON-Bereitstellungen Vorrang haben.

3. Führen Sie die Attribute der Stack-Konfiguration und der Bereitstellung in das Kontenobjekt der Instance ein.
4. Führen Sie die Attribute der in die Instances integrierten Rezeptbücher in das Knotenobjekt ein.

   Die Attribute der integrierten Rezeptbücher sind alle `default`-Typen. Wenn ein oder mehrere integrierte Cookbook-Attribute auch in der Stack-Konfiguration und in den Bereitstellungsattributen definiert sind — in der Regel, weil Sie sie mit benutzerdefiniertem JSON definiert haben —, haben die Definitionen der Stack-Konfiguration Vorrang vor den integrierten Cookbook-Definitionen. Alle anderen integrierten Rezeptbuchattribute werden einfach in das Knotenobjekt integriert.
5. Führen Sie die Attribute der benutzerdefinierten Rezeptbücher der Instances in das Knotenobjekt ein.

   Attribute von benutzerdefinierten Rezeptbüchern sind in der Regel entweder `normal`- oder `default`-Typen. Einmalige Attribute werden in das Knotenobjekt integriert. Wenn in den Schritten 1—3 auch benutzerdefinierte Kochbuchattribute definiert werden (normalerweise, weil Sie sie mit benutzerdefiniertem JSON definiert haben), hängt die Rangfolge vom Typ des benutzerdefinierten Kochbuchattributs ab:

   - In den Schritten 1—3 definierte Attribute haben Vorrang vor benutzerdefinierten Kochbuchattributen. `default`
   - Benutzerdefinierte `normal` Kochbuchattribute haben Vorrang vor Definitionen aus den Schritten 1—3.

> **Important**
>
> Verwenden Sie keine benutzerdefinierten Rezeptbuchattribute vom `default`-Typ, um Attribute der Stack-Konfiguration oder eines integrierten Rezeptbuches zu überschreiben. Da benutzerdefinierte Rezeptbuchattribute zuletzt evaluiert werden, sind diese `default`-Attribute die letzten in der Rangfolge und können nichts überschreiben.

Überschreiben von Attributen mit einem benutzerdefinierten JSON-Objekt

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Da AWS OpsWorks Stacks Chef-Läufe für Windows-Stacks anders handhabt als für Linux-Stacks, können Sie die in diesem Abschnitt beschriebenen Techniken nicht für Windows-Stacks verwenden.

Die einfachste Möglichkeit, ein AWS OpsWorks Stacks-Attribut zu überschreiben, besteht darin, es in benutzerdefiniertem JSON zu definieren, das Vorrang vor Stackkonfigurations- und Bereitstellungsattributen sowie integrierten und benutzerdefinierten Cookbook-Attributen hat. `default` Weitere Informationen finden Sie unter Priorität von Attributen.

> **Important**
>
> Sie sollten die Attribute der Stack-Konfiguration und Bereitstellung mit Bedacht überschreiben. Beispielsweise kann das Überschreiben von Attributen im `opsworks`-Namespace integrierte Rezepte stören. Weitere Informationen finden Sie unter Attribute für die Stack-Konfiguration und -Bereitstellung.

Sie können auch ein benutzerdefiniertes JSON-Objekt verwenden, um eindeutige Attribute zur Übertragung von Daten an Ihre benutzerdefinierten Rezepte zu definieren. Die Attribute werden einfach in das Knotenobjekt integriert und Rezepte können mithilfe der standardmäßigen Chef-Knotensyntax auf diese verweisen.

Angeben eines benutzerdefinierten JSON-Objekts

Um mit einem benutzerdefinierten JSON-Objekt einen Attributwert zu überschreiben, müssen Sie zuerst den vollständig qualifizierten Attributnamen dieses Attributs ermitteln. Anschließend erstellen Sie ein JSON-Objekt mit den Attributen, die Sie überschreiben möchten, eingestellt auf Ihre bevorzugten Werte. Zur Vereinfachung nutzen die Dokumente Stack-Konfigurations- und Bereitstellungsattribute: Linux und Integrierte Rezeptbuchattribute normalerweise Stack-Konfigurations-, Bereitstellungs- und integrierte Rezeptbuchattribute, einschließlich ihrer vollständig qualifizierten Namen.

Die Beziehungen zwischen über- und untergeordneten Elementen des Objekts müssen mit den entsprechenden vollständig qualifizierten Chef-Knoten übereinstimmen. Angenommen Sie möchten z. B. die folgenden Apache-Attribute ändern:

- Das `keepalivetimeout`-Attribut, dessen Knoten `node[:apache][:keepalivetimeout]` ist und das einen Standardwert von `3` hat.
- Das `logrotate` `schedule`-Attribut, dessen Knoten `node[:apache][:logrotate][:schedule]` ist und hat einen Standardwert von `"daily"` hat.

Um die Attribute zu überschreiben und die Werte auf `5` und `"weekly"` festzulegen, nutzen Sie das folgende benutzerdefinierte JSON-Objekt:

```
{
  "apache" : {
    "keepalivetimeout" : 5,
    "logrotate" : {
       "schedule" : "weekly"
    }
  }
}
```

Angeben des benutzerdefinierten JSON-Objekts zum richtigen Zeitpunkt

Sie können eine benutzerdefinierte JSON-Struktur für die folgenden Aufgaben angeben:

- Einen neuen Stack erstellen
- Einen Stack aktualisieren
- Einen Stack-Befehl ausführen

- Einen Stack klonen
- Bereitstellen einer Anwendung

Für jede Aufgabe führt AWS OpsWorks Stacks die benutzerdefinierten JSON-Attribute mit den Stackkonfigurations- und Bereitstellungsattributen zusammen und sendet sie an die Instanzen, damit sie mit dem Node-Objekt zusammengeführt werden. Beachten Sie jedoch Folgendes:

- Wenn Sie ein benutzerdefiniertes JSON-Objekt beim Erstellen, Klonen oder Aktualisieren eines Stacks angeben, werden die Attribute in die Attribute der Stack-Konfiguration und der Bereitstellung für alle nachfolgenden Lebenszyklusereignisse und Stack-Befehle eingeführt.
- Wenn Sie ein benutzerdefiniertes JSON-Objekt für eine Bereitstellung angeben, werden die Attribute nur für das entsprechende Ereignis in die Attribute der Stack-Konfiguration und der Bereitstellung eingeführt.

  Wenn Sie diese benutzerdefinierten Attribute für nachfolgende Bereitstellungen verwenden möchten, müssen Sie das benutzerdefinierte JSON-Objekt noch einmal explizit angeben.

Beachten Sie, dass Attribute nur Auswirkungen auf die Instance haben, wenn sie von Rezepten verwendet werden. Wenn Sie einen Attributwert überschreiben, aber keine nachfolgenden Rezepte auf dieses Attribut zurückgreifen, hat die Änderung keine Auswirkungen. Sie müssen entweder sicherstellen, dass das benutzerdefinierte JSON-Objekt gesendet wurde, bevor die damit in Verbindung stehenden Rezepte ausgeführt werden, oder dass die in Verbindung stehenden Rezepte noch einmal ausgeführt werden.

Bewährte Methoden für die Verwendung eines benutzerdefinierten JSON-Objekts

Sie können benutzerdefiniertes JSON verwenden, um jedes AWS OpsWorks Stacks-Attribut zu überschreiben, aber die manuelle Eingabe der Informationen ist etwas umständlich und unterliegt keiner Quellcodeverwaltung. Ein benutzerdefiniertes JSON-Objekt eignet sich am besten für folgende Situationen:

- Wenn Sie nur eine kleine Anzahl von Attributen überschreiben möchten und Sie nicht anderweitig auf die Verwendung benutzerdefinierter Rezeptbücher angewiesen sind.

  Durch die Verwendung eines benutzerdefinierten JSON-Objekts können Sie den Aufwand beim Einrichten und Warten eines Rezeptbuch-Repositorys, nur um ein paar Attribute zu überschreiben, vermeiden.

- Sensible Daten, wie z. B. Passwörter oder Authentifizierungsschlüssel.

  Rezeptbuchattribute werden in einem Repository gespeichert, so dass alle vertraulichen Informationen einem gewissen Risiko ausgesetzt sind, kompromittiert zu werden. Definieren Sie Attribute stattdessen mit Dummy-Werten und benutzen Sie das benutzerdefinierte JSON-Objekt, um die tatsächlichen Werte anzugeben.

- Werte, die erfahrungsgemäß abweichen.

  Ein empfohlenes Verfahren ist beispielsweise, dass Ihre Produktion den Stack durch separate Entwicklungs- und Staging-Stacks unterstützt. Angenommen, diese Stacks unterstützen eine Anwendung, die Zahlungen akzeptiert. Wenn Sie ein benutzerdefiniertes JSON-Objekt nutzen, um den Endpunkt der Zahlung anzugeben, können Sie eine Test-URL für Ihren Staging-Stack angeben. Wenn Sie bereit sind, einen aktualisierten Stack zu Ihrem Produktions-Stack zu migrieren, können Sie die gleichen Rezeptbücher und ein benutzerdefiniertes JSON-Objekt benutzen, um den Endpunkt der Zahlung auf die Produktions-URL zu setzen.

- Werte, die einem bestimmten Stack- oder Bereitstellungsbefehl zugeordnet sind.

Überschreiben von AWS OpsWorks Stacks-Attributen mithilfe von benutzerdefinierten Cookbook-Attributen

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Für Windows-Stacks verwendet AWS OpsWorks Stacks separate Chef-Läufe für integrierte Rezepte und benutzerdefinierte Rezepte. Dies bedeutet, dass Sie die in diesem Abschnitt erläuterten Techniken nicht verwenden können, um integrierte Attribute für Windows-Stacks zu überschreiben.

Benutzerdefiniertes JSON ist eine bequeme Möglichkeit, die AWS OpsWorks Stacks-Stack-Konfiguration und die integrierten Cookbook-Attribute zu überschreiben, hat jedoch einige Einschränkungen. Sie müssen vor allem das benutzerdefinierte JSON-Objekt manuell für jede Nutzung eingeben. Somit haben Sie keine robuste Methode zum Verwalten der Definitionen. Ein besserer Ansatz ist häufig die Verwendung von Dateien benutzerdefinierter Rezeptbuchattribute, um integrierte Attribute zu überschreiben. Auf diese Weise können Sie die Definitionen in einer Quellüberwachung platzieren.

Das Verfahren zur Verwendung von Dateien mit benutzerdefinierten Attributen zum Überschreiben von AWS OpsWorks Stacks-Definitionen ist unkompliziert.

Um AWS OpsWorks Stacks-Attributdefinitionen zu überschreiben

1. Richten Sie ein Rezeptbuch-Repository ein, wie in Cookbooks und Rezepte beschrieben.
2. Erstellen Sie ein Rezeptbuch mit demselben Namen wie das integrierte Rezeptbuch, das die Attribute enthält, die Sie überschreiben möchten. Um beispielsweise die Apache-Attribute zu überschreiben, sollte das Rezeptbuch "apache2" genannt werden.
3. Fügen Sie den Ordner `attributes` zum Rezeptbuch hinzu und fügen Sie diesem Ordner eine Datei mit dem Namen `customize.rb` hinzu.
4. Fügen Sie der Datei eine Attributdefinition für jede der integrierten Rezeptbuchattribute hinzu, die Sie überschreiben möchten. Stellen Sie diese auf Ihren bevorzugten Wert ein. Das Attribut muss einen `normal` Typ oder höher haben und genau denselben Knotennamen wie das entsprechende AWS OpsWorks Stacks-Attribut haben. Eine ausführliche Liste der AWS OpsWorks Stacks-Attribute, einschließlich Knotennamen, finden Sie unter Stack-Konfigurations- und Bereitstellungsattribute: Linux und. Integrierte Rezeptbuchattribute Weitere Informationen zu Attributen und Attributdateien finden Sie unter About Attribute Files.

> **Important**
>
> Ihre Attribute müssen `normal` vom Typ sein, um AWS OpsWorks Stacks-Attribute zu überschreiben. `default` Typen haben keinen Vorrang. Wenn Ihre Datei `customize.rb` z. B. eine `default[:apache][:keepalivetimeout] = 5`-Attributdefinition enthält, wird das entsprechenden Attribut in der integrierten Attributdatei `apache.rb` zuerst berechnet und hat Vorrang. Weitere Informationen finden Sie unter Überschreiben der Attribute.

5. Wiederholen Sie die Schritte 2 bis 4 für jedes integrierte Kochbuch mit Attributen, die Sie überschreiben möchten.
6. Aktivieren Sie benutzerdefinierte Kochbücher für Ihren Stack und geben Sie die Informationen an, die AWS OpsWorks Stacks benötigt, um Ihre Kochbücher auf die Instanzen des Stacks herunterzuladen. Weitere Informationen finden Sie unter Installieren von benutzerdefinierten Rezeptbüchern.

> **Note**
>
> Eine komplette schrittweise Anleitung dieses Verfahrens finden Sie unter Überschreiben von integrierten Attributen.

Das Knotenobjekt, das von nachfolgenden Lebenszyklusereignissen, Deploy-Befehlen und Stack-Befehlen verwendet wird, enthält jetzt Ihre Attributdefinitionen anstelle der AWS OpsWorks Stacks-Werte.

Zum Überschreiben der integrierten Apache-Einstellungen `keepalivetimeout` und `logrotate schedule`, welche in Angeben eines benutzerdefinierten JSON-Objekts erläutert wurden, fügen Sie beispielsweise ein `apache2`-Rezeptbuch zu Ihrem Repository hinzu und eine `customize.rb`-Datei zum `attributes`-Rezeptbuchordner mit den folgenden Inhalten.

```
normal[:apache][:keepalivetimeout] = 5
normal[:apache][:logrotate][:schedule] = 'weekly'
```

> **Important**
>
> Sie sollten AWS OpsWorks Stacks-Attribute nicht überschreiben, indem Sie eine Kopie der zugehörigen integrierten Attributdatei ändern. Wenn Sie beispielsweise `apache.rb` in Ihren Ordner `apache2/attributes` kopieren und einige seiner Einstellungen ändern, überschreiben Sie im Wesentlichen jedes Attribut in der integrierten Datei. Die Rezepte verwenden die Attributdefinitionen von Ihrer Kopie und ignorieren die integrierte Datei. Wenn AWS OpsWorks Stacks zu einem späteren Zeitpunkt die integrierte Attributdatei ändert, haben die Rezepte keinen Zugriff auf die Änderungen, es sei denn, Sie aktualisieren Ihre Kopie manuell.

Um dies zu verhindern, enthalten alle integrierten Rezeptbücher eine leere `customize.rb`-Attributedatei, die in allen Modulen durch eine `include_attribute`-Richtlinie erforderlich ist. Durch Überschreiben der Attribute in Ihrer Kopie `customize.rb` beeinflussen Sie nur die spezifischen Attribute. Die Rezepte beschaffen jegliche anderen Attributwerte aus den integrierten Attributdateien und bekommen automatisch die aktuellen Werte aller Attribute, die Sie nicht überschrieben haben.
Mit diesem Ansatz können Sie die Anzahl der Attribute in Ihrem Rezeptbuch-Repository klein halten, wodurch Ihre Fixkosten für die Wartung gering gehalten werden und zukünftige Upgrades einfacher zu verwalten sind.

Erweitern von AWS OpsWorks Stacks-Konfigurationsdateien mithilfe benutzerdefinierter Vorlagen

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Da AWS OpsWorks Stacks Chef-Läufe für Windows-Stacks anders handhabt als für Linux-Stacks, können Sie die in diesem Abschnitt beschriebenen Techniken nicht für Windows-Stacks verwenden.

AWS OpsWorks Stacks verwendet Vorlagen, um Dateien wie Konfigurationsdateien zu erstellen, die in der Regel von Attributen für viele Einstellungen abhängen. Wenn Sie benutzerdefinierte JSON- oder benutzerdefinierte Cookbook-Attribute verwenden, um die AWS OpsWorks Stacks-Definitionen zu überschreiben, werden Ihre bevorzugten Einstellungen anstelle der Stacks-Einstellungen in die Konfigurationsdateien aufgenommen. AWS OpsWorks AWS OpsWorks Stacks spezifiziert jedoch nicht unbedingt ein Attribut für jede mögliche Konfigurationseinstellung; es akzeptiert die Standardeinstellungen für einige Einstellungen und codiert andere direkt in der Vorlage fest. Sie können keine benutzerdefinierten JSON- oder benutzerdefinierten Kochbuchattribute verwenden,

um bevorzugte Einstellungen anzugeben, wenn es kein entsprechendes Stacks-Attribut gibt. AWS OpsWorks

Sie können die Konfigurationsdatei erweitern, um zusätzliche Konfigurationseinstellungen aufzunehmen, indem Sie eine benutzerdefinierte Vorlage erstellen. Anschließend können Sie beliebige Konfigurationseinstellungen oder andere erforderliche Inhalte zur Datei hinzufügen und alle fest programmierten Einstellungen überschreiben. Weitere Informationen zu Vorlagen finden Sie unter Vorlagen.

> **Note**
>
> Sie können alle integrierten Vorlagen mit Ausnahme von opsworks-agent.monitrc.erb überschreiben.

So erstellen Sie eine benutzerdefinierte Vorlage

1. Erstellen Sie ein Rezeptbuch mit derselben Struktur und denselben Verzeichnisnamen wie das integrierte Rezeptbuch. Erstellen Sie dann im entsprechenden Verzeichnis eine Vorlagendatei mit demselben Namen wie die integrierte Vorlage, die Sie anpassen möchten. Wenn Sie beispielsweise eine benutzerdefinierte Vorlage zum Erweitern der Apache-Konfigurationsdatei `httpd.conf` verwenden, müssen Sie ein `apache2`-Rezeptbuch in Ihrem Repository implementieren und Ihre Vorlagendatei `apache2/templates/default/apache.conf.erb` nennen. Wenn Sie genau dieselben Namen verwenden, kann AWS OpsWorks Stacks die benutzerdefinierte Vorlage erkennen und sie anstelle der integrierten Vorlage verwenden.

   Der einfachste Ansatz besteht darin, einfach die integrierte Vorlagendatei aus dem GitHubRepository des integrierten Kochbuchs in Ihr Kochbuch zu kopieren und sie nach Bedarf zu ändern.

   > **Important**
   >
   > Kopieren Sie keine Dateien aus dem integrierten Rezeptbuch, mit Ausnahme der anzupassenden Vorlagendateien. Durch Kopieren anderer Arten von Rezeptbuch-Dateien, wie z. B. Rezepte, werden doppelte Chef-Ressourcen erstellt und es können Fehler auftreten.

Das Rezeptbuch kann auch benutzerdefinierte Attribute, Rezepte und zugehörige Dateien enthalten, jedoch sollten deren Dateinamen keine doppelten Namen integrierter Dateien enthalten.

2. Passen Sie die Vorlagendatei an, um eine Konfigurationsdatei entsprechend Ihren Anforderungen zu erstellen. Sie können weitere Einstellungen hinzufügen, vorhandene Einstellungen löschen, fest programmierte Attribute ersetzen usw.
3. Sofern Sie es nicht bereits getan haben, bearbeiten Sie die Stack-Einstellungen, um benutzerdefinierte Rezeptbücher zu aktivieren, und legen Sie Ihr Rezeptbuch-Repository fest. Weitere Informationen finden Sie unter Installieren von benutzerdefinierten Rezeptbüchern.

> **Note**
>
> Eine komplette schrittweise Anleitung dieses Verfahrens finden Sie unter Überschreiben von integrierten Vorlagen.

Sie müssen keine Rezepte implementieren oder Rezepte zur Layer-Konfiguration hinzufügen, um eine Vorlage zu überschreiben. AWS OpsWorks Stacks führt immer die integrierten Rezepte aus. Bei der Ausführung des Rezepts, mit dem die Konfigurationsdatei erstellt wird, wird Ihre benutzerdefinierte Vorlage automatisch anstelle der integrierten Vorlage verwendet.

> **Note**
>
> Wenn AWS OpsWorks Stacks Änderungen an der integrierten Vorlage vornimmt, ist Ihre benutzerdefinierte Vorlage möglicherweise nicht mehr synchron und funktioniert nicht mehr richtig. Nehmen wir zum Beispiel an, Ihre Vorlage bezieht sich auf eine abhängige Datei und der Dateiname ändert sich. AWS OpsWorks Stacks nimmt solche Änderungen nicht oft vor, und wenn sich eine Vorlage ändert, listet es die Änderungen auf und gibt Ihnen die Möglichkeit, auf eine neue Version zu aktualisieren. Sie sollten das AWS OpsWorks Stacks-Repository auf Änderungen überwachen und Ihre Vorlage bei Bedarf manuell aktualisieren.

Erweitern eines Layers

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Es gibt Fälle, in denen Sie einen integrierten Layer über das Maß hinaus anpassen müssen, das durch Bearbeiten der AWS OpsWorks Stacks-Attribute oder Anpassen von Vorlagen möglich ist. Angenommen, Sie müssen symbolische Links erstellen, Datei- oder Verzeichnis-Modi festlegen oder zusätzliche Pakete installieren. Sie müssen benutzerdefinierte Layer erweitern, um weitere Funktionen hinzuzufügen. In diesem Fall müssen Sie mindestens ein benutzerdefiniertes Rezeptbuch mit Rezepten für die Anpassung implementieren. In diesem Thema finden Sie einige Beispiele dafür, wie Sie mit Rezepten Layer erweitern können.

Wenn Sie noch nie mit Chef gearbeitet haben, lesen Sie zuerst das Tutorial Rezeptbücher 101. Es enthält eine Einführung in die Grundlagen der Implementierung von Rezeptbüchern, mit denen Sie die unterschiedlichsten Aufgaben ausführen können. Ein detailliertes Beispiel für die Implementierung eines benutzerdefinierten Layers finden Sie unter Erstellen eines benutzerdefinierten Tomcat-Server-Layers.

Themen

- Verwenden von Rezepten zum Ausführen von Skripten
- Verwenden von Chef-Bereitstellungs-Hooks
- Ausführen von Cron-Jobs auf Linux-Instances
- Installieren und Konfigurieren von Paketen auf Linux-Instances

Verwenden von Rezepten zum Ausführen von Skripten

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir

empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn Sie bereits über ein Skript verfügen, das die notwendigen Anpassungen vornehmen kann, ist es oft am einfachsten, zur Erweiterung des Layers ein Rezept zu implementieren, um ein Skript auszuführen. Dieses Rezept können Sie dann dem passenden Lebenszyklusereignis, in der Regel Einrichtung oder Bereitstellung, zuweisen oder das Rezept mit dem Stack-Befehl `execute_recipes` manuell ausführen.

Im folgenden Beispiel wird ein Shell-Skript auf Linux-Instances ausgeführt. Sie können den gleichen Ansatz jedoch auch für andere Skripttypen verwenden, einschließlich PowerShell Windows-Skripts.

```
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
```

Die Ressource `cookbook_file` repräsentiert eine Datei, die in einem Unterverzeichnis des Verzeichnisses `files` eines Rezeptbuchs gespeichert ist, und die die Datei an einen festgelegten Ort auf der Instance überträgt. In diesem Beispiel wird ein Shell-Skript, `lib-installer.sh`, in das Verzeichnis `/tmp` der Instance übertragen und der Dateimodus auf `0755` gesetzt. Weitere Informationen finden Sie unter cookbook_file.

Die Ressource `execute` steht für einen Befehl, beispielsweise einen Shell-Befehl. In diesem Beispiel wird `lib-installer.sh` ausgeführt. Weitere Informationen finden Sie unter execute.

Sie können Skripte auch ausführen, indem Sie sie in ein Rezept aufnehmen. Im folgenden Beispiel wird ein Bash-Skript ausgeführt. Chef unterstützt jedoch auch Csh, Perl, Python und Ruby.

```
script "install_something" do
  interpreter "bash"
```

```
  user "root"
  cwd "/tmp"
  code <<-EOH
    #insert bash script
  EOH
end
```

Die Ressource `script` repräsentiert ein Skript. In diesem Beispiel wird ein Bash-Interpreter festgelegt, der Benutzer ist `"root"` und das Arbeitsverzeichnis ist `/tmp`. Dann wird das Bash-Skript im `code`-Block ausgeführt, der beliebig viele Zeilen enthalten kann. Weitere Informationen finden Sie unter script.

Weitere Informationen dazu, wie Sie mithilfe von Rezepten Skripte ausführen, finden Sie unter Beispiel 7: Ausführen von Befehlen und Skripts. Ein Beispiel für die Ausführung eines PowerShell Skripts auf einer Windows-Instanz finden Sie unterEin PowerShell Windows-Skript ausführen.

Verwenden von Chef-Bereitstellungs-Hooks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können die Bereitstellung anpassen, indem Sie ein benutzerdefiniertes Rezept für die erforderlichen Aufgaben implementieren und es dem passenden Bereitstellungsereignis des Layers zuweisen. Ein alternativer und manchmal einfacherer Ansatz — insbesondere, wenn Sie kein Kochbuch für andere Zwecke implementieren müssen — besteht darin, Chef-Bereitstellungs-Hooks zu verwenden, um Ihren Anpassungscode auszuführen. Außerdem werden benutzerdefinierte Bereitstellungsrezepte nach der Bereitstellung durch integrierte Rezepte ausgeführt. Mithilfe von Bereitstellungs-Hooks können Sie in eine Bereitstellung eingreifen, beispielsweise nachdem der App-Code aus dem Repository abgerufen wurde, aber bevor Apache neu gestartet wird.

Chef stellt Apps in vier Phasen bereit:

- Checkout — Lädt die Dateien aus dem Repository herunter

- Migrieren — Führt bei Bedarf eine Migration durch
- Symlink — Erzeugt Symlinks
- Restart — Startet die Anwendung neu

Chef-Bereitstellungs-Hooks sind eine einfache Möglichkeit, eine Bereitstellung anzupassen, indem Sie optional nach Abschluss einer Phase eine vom Benutzer erstellte Ruby-Anwendung ausführen können. Um Bereitstellungs-Hooks zu verwenden, implementieren Sie mindestens eine Ruby-Anwendung und speichern diese im Verzeichnis `/deploy` Ihrer App. (Wenn das Verzeichnis `/deploy` noch nicht vorhanden ist, erstellen Sie es auf `APP_ROOT`-Ebene.) Die Anwendung muss einen der folgenden Namen haben, über den festgelegt wird, wann sie ausgeführt wird.

- `before_migrate.rb` wird nach der Checkout-Phase, aber vor der Migrationsphase ausgeführt.
- `before_symlink.rb` wird nach der Migrationsphase, aber vor der Symlink-Phase ausgeführt.
- `before_restart.rb` wird nach der Symlink-Phase, aber vor der Neustartphase ausgeführt.
- `after_restart.rb` wird nach der Neustartphase ausgeführt.

Chef-Bereitstellungs-Hooks können genau wie Rezepte über die Standard-Knotensyntax auf das Knotenobjekt zugreifen. Außerdem können Bereitstellungs-Hooks auf die Werte beliebiger App-Umgebungsvariablen zugreifen, die Sie festgelegt haben. Sie müssen jedoch `new_resource.environment["VARIABLE_NAME"]` anstelle von `ENV["VARIABLE_NAME"]` verwenden, um auf die Variablenwerte zuzugreifen.

Ausführen von Cron-Jobs auf Linux-Instances

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Ein Cron-Job unter Linux weist den Cron-Daemon an, Befehle zu festgelegten Zeiten auszuführen. Angenommen, Ihr Stack unterstützt eine PHP-E-Commerce-Anwendung. Sie können einen Cron-Job einrichten, um den Server anzuweisen, wöchentlich zu einem festgelegten Zeitpunkt einen

Verkaufsbericht zu senden. Weitere Informationen zu Cron finden Sie unter Cron auf Wikipedia. Weitere Informationen dazu, wie Sie einen Cron-Auftrag direkt auf einem Linux-basierten Computer oder einer Linux-basierten Instance ausführen, finden Sie unter Was ist Cron und Crontab und wie sind Sie zu verwenden? auf der Wissensdatenbank-Website der Indiana University.

Sie können `cron`-Jobs zwar auf einzelnen Linux-basierten Instances manuell einrichten, indem Sie sich über SSH auf den Instances anmelden und ihre `crontab`-Einträge, bearbeiten, ein großer Vorteil von AWS OpsWorks Stacks besteht jedoch darin, dass Sie damit den Auftrag für einen ganze Instances-Layer ausführen können. Das folgende Verfahren beschreibt, wie Sie einen `cron` Job auf den Instanzen einer PHP App Server-Layer einrichten, aber Sie können den gleichen Ansatz für jede Ebene verwenden.

So richten Sie einen **`cron`**-Job auf den Instances eines Layers ein

1. Implementieren Sie ein Kochrezept mit einem Rezept, das eine `cron`-Ressource enthält, um den Auftrag einzurichten. In diesem Beispiel heißt das Rezept `cronjob.rb`. Weitere Informationen zur Implementierung werden im weiteren Verlauf dieser Anleitung beschrieben. Weitere Informationen zu Rezeptbüchern und Rezepten finden Sie unter Cookbooks und Rezepte.
2. Installieren Sie das Rezeptbuch auf Ihrem Stack. Weitere Informationen finden Sie unter Installieren von benutzerdefinierten Rezeptbüchern.
3. Lassen Sie AWS OpsWorks Stacks das Rezept automatisch auf den Instanzen der Ebene ausführen, indem Sie es den folgenden Lebenszyklusereignissen zuweisen. Weitere Informationen finden Sie unter Automatisches Ausführen von Rezepten.

   - Setup — Durch die Zuweisung `cronjob.rb` zu diesem Ereignis wird AWS OpsWorks Stacks angewiesen, das Rezept auf allen neuen Instanzen auszuführen.
   - Bereitstellen — Durch die Zuweisung `cronjob.rb` zu diesem Ereignis wird AWS OpsWorks Stacks angewiesen, das Rezept auf allen Online-Instanzen auszuführen, wenn Sie eine App auf dem Layer bereitstellen oder erneut bereitstellen.

   Sie können das Rezept auch manuell auf Online-Instances ausführen. Verwenden Sie dazu den Stack-Befehl `Execute Recipes`. Weitere Informationen finden Sie unter Ausführen von Stack-Befehlen.

Nachfolgend finden Sie das Beispiel `cronjob.rb`, mit dem ein Cron-Job eingerichtet wird, um einmal wöchentlich eine vom Benutzer implementierte PHP-Anwendung auszuführen, die

Verkaufsdaten vom Server abruft und einen Bericht per E-Mail sendet. Weitere Informationen zur Verwendung von Cron-Ressourcen finden Sie unter cron.

```
cron "job_name" do
  hour "1"
  minute "10"
  weekday "6"
  command "cd /srv/www/myapp/current && php .lib/mailing.php"
end
```

`cron` ist eine Chef-Ressource, die einen `cron`-Auftrag repräsentiert. Wenn AWS OpsWorks Stacks das Rezept auf einer Instanz ausführt, kümmert sich der zugehörige Anbieter um die Details der Einrichtung des Jobs.

- *`job_name`* ist ein benutzerdefinierter Name für den `cron`-Auftrag, beispielsweise `weekly report`.
- Über `hour`/`minute`/`weekday` legen Sie den Zeitpunkt fest, zu dem die Befehle ausgeführt werden. In diesem Beispiel werden die Befehle samstags um 1.10 Uhr ausgeführt.
- `command` legt die auszuführenden Befehle fest.

  In diesem Beispiel werden zwei Befehle ausgeführt. Der erste Befehl führt zum Verzeichnis `/srv/www/myapp/current`. Der zweite Befehl führt die vom Benutzer implementierte Anwendung `mailing.php` aus, über die Verkaufsdaten erfasst und als Bericht gesendet werden.

> **Note**
>
> Der Befehl `bundle` ist standardmäßig nicht mit `cron`-Aufträgen kompatibel. Der Grund dafür ist, dass AWS OpsWorks Stacks den Bundler im Verzeichnis installiert. `/usr/local/bin` Um den Befehl `bundle` in einem `cron`-Auftrag zu verwenden, müssen Sie den Pfad `/usr/local/bin` dem Cron-Auftrag hinzufügen. Da auch die Umgebungsvariable $PATH sich möglicherweise nicht auf den `cron`-Auftrag auswirkt, sollten Sie alle erforderlichen Pfadinformationen explizit zu dem Auftrag hinzufügen und sich nicht darauf verlassen, dass die $PATH-Variable sich automatisch auf den Cron-Auftrag auswirkt. In den folgenden Beispielen lernen Sie zwei Möglichkeiten zur Verwendung von `bundle` in einem `cron`-Auftrag kennen.

```
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
```

Wenn Ihr Stack über mehrere Anwendungsserver verfügt, ist die `cronjob.rb` Zuweisung von Ereignissen im Lebenszyklus der PHP App Server-Ebene möglicherweise kein idealer Ansatz. Wenn das Rezept auf allen Instances des Layers ausgeführt wird, erhalten Sie beispielsweise mehrere Berichte. Verwenden Sie besser einen benutzerdefinierten Layer, um sicherzustellen, dass nur von einem Server ein Bericht gesendet wird.

So führen Sie ein Rezept auf nur einer Instance eines Layers aus

1. Erstellen Sie einen benutzerdefinierten Layer und nennen Sie ihn beispielsweise PHPAdmin. Weisen Sie `cronjob.rb` den Lebenszyklusereignissen Einrichtung und Bereitstellung zu. Benutzerdefinierte Layer müssen nicht immer viel tun. In diesem Fall führt PHPAdmin nur ein benutzerdefiniertes Rezept auf seinen Instances aus.
2. Weisen Sie eine der PHP App Server-Instanzen zu AdminLayer. Wenn eine Instanz zu mehr als einer Ebene gehört, führt AWS OpsWorks Stacks die integrierten und benutzerdefinierten Rezepte jeder Ebene aus.

Da nur eine Instanz zu den Ebenen PHP App Server und phpAdmin gehört, `cronjob.rb` wird sie nur auf dieser Instanz ausgeführt und Sie erhalten nur einen Bericht.

Installieren und Konfigurieren von Paketen auf Linux-Instances

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Die integrierten Layer unterstützen nur bestimmte Pakete. Weitere Informationen finden Sie unter Ebenen. Um andere Pakete wie einen Redis-Server zu installieren, müssen Sie benutzerdefinierte Rezepte für Einrichtung, Konfiguration und Bereitstellung implementieren. Es gibt Fälle, in denen es am besten ist, einen integrierten Layer zu erweitern, um das Paket zusätzlich zu den Standardpaketen des Layers auf den Instances des Layers zu installieren. Wenn Sie beispielsweise über einen Stack verfügen, der eine PHP-Anwendung unterstützt, und Sie einen Redis-Server hinzufügen möchten, können Sie die PHP App Server-Ebene erweitern, um zusätzlich zu einem PHP-Anwendungsserver auch einen Redis-Server auf den Instanzen der Ebene zu installieren und zu konfigurieren.

Ein Paketinstallationsrezept muss in der Regel folgende Aufgaben ausführen:

- Erstellen von Verzeichnissen und Festlegen von deren Modi
- Erstellen einer Konfigurationsdatei aus einer Vorlage
- Ausführen des Installationsprogramms, um das Paket auf der Instance zu installieren
- Starten von Services

Ein Beispiel für die Installation eines Tomcat-Servers finden Sie unter Erstellen eines benutzerdefinierten Tomcat-Server-Layers. In diesem Thema wird beschrieben, wie Sie einen benutzerdefinierten Redis-Layer einrichten. Mit nahezu demselben Code können Sie Redis jedoch auch auf einem integrierten Layer installieren und konfigurieren. Beispiele für die Installation anderer Pakete finden Sie in den integrierten Kochbüchern unter https://github.com/aws/opsworks-cookbooks.

Erstellen eines benutzerdefinierten Tomcat-Server-Layers

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> In diesem Thema wird beschrieben, wie ein benutzerdefinierter Layer für einen Linux-Stack implementiert wird. Die grundlegenden Prinzipien und ein Teil des Codes kann jedoch auch zur Implementierung benutzerdefinierter Layer für Windows-Stacks angepasst werden, vor allem diejenigen im Abschnitt über Anwendungsbereitstellung.

Die einfachste Möglichkeit, nicht standardmäßige Pakete auf AWS OpsWorks Stacks-Instanzen zu verwenden, besteht darin, eine bestehende Ebene zu erweitern. Bei diesem Ansatz werden jedoch sowohl standardmäßige als auch nicht standardmäßige Pakete auf Instances des Layers installiert und ausgeführt, was nicht immer wünschenswert ist. Ein etwas komplexerer, aber leistungsstärkerer Ansatz besteht darin, einen benutzerdefinierten Layer zu implementieren. Dies gibt Ihnen die nahezu vollständige Kontrolle über die Instances des Layers, einschließlich der folgenden Aspekte:

- Welche Pakete installiert werden
- Wie jedes Paket konfiguriert ist
- Wie Anwendungen von einem Repository der Instance bereitgestellt werden

Unabhängig davon, ob Sie die Konsole oder die API verwenden, erstellen und verwalten Sie einen benutzerdefinierten Layer ähnlich wie jeden anderen Layer, wie unter Benutzerspezifische Layers beschrieben. Die integrierten Rezepte eines benutzerdefinierten Layers führen jedoch nur einige sehr grundlegende Aufgaben aus, wie z. B. das Installieren eines Ganglia-Clients zum Melden von Metriken an einen Ganglia-Master. Um die Instances eines benutzerdefinierten Layers mehr als minimal funktionsfähig zu machen, müssen Sie mindestens ein benutzerdefiniertes Rezeptbuch mit Chef-Rezepten und zugehörige Dateien implementieren, um die Aufgaben der Installation und

Konfiguration von Paketen, der Bereitstellung von Anwendungen usw. auszuführen. Aber Sie müssen nicht notwendigerweise alles von Grund auf implementieren. Wenn Sie beispielsweise Anwendungen in einem der Standard-Repositorys speichern, können Sie die integrierten Bereitstellungsrezepte verwenden, um einen Großteil der Arbeit der Installation der Anwendungen auf den Instances des Layers zu verarbeiten.

> **Note**
>
> Wenn Sie noch nie mit Chef gearbeitet haben, lesen Sie zuerst das Tutorial Rezeptbücher 101. Es enthält eine Einführung in die Grundlagen der Implementierung von Rezeptbüchern, mit denen Sie die unterschiedlichsten Aufgaben ausführen können.

In der folgenden schrittweisen Anleitung wird beschrieben, wie Sie einen benutzerdefinierten Layer implementieren, der einen Tomcat-Anwendungsserver unterstützt. Der Layer basiert auf einem benutzerdefinierten Rezeptbuch mit dem Namen Tomcat. Es enthält Rezepte zum Verarbeiten von Paketinstallation, Bereitstellung usw. Die schrittweise Anleitung umfasst Auszüge aus dem Tomcat-Rezeptbuch. Sie können das komplette Kochbuch aus dem Repository herunterladen. GitHub Wenn Sie mit Opscode Chef nicht vertraut sind, sollten Sie zunächst Cookbooks und Rezepte lesen.

> **Note**
>
> AWS OpsWorks Stacks enthält eine Java App Server-Schicht mit vollem Funktionsumfang für den Produktionseinsatz. Der Zweck des Tomcat-Rezeptbuchs besteht darin zu zeigen, wie benutzerdefinierte Layer implementiert werden. Es unterstützt daher nur eine eingeschränkte Version von Tomcat, die keine Funktionen wie SSL umfasst. Ein Beispiel für eine Implementierung mit vollem Funktionsumfang finden Sie in dem integrierten Rezeptbuch opsworks_java.

Das Tomcat-Rezeptbuch unterstützt einen benutzerspezifischen Layer, dessen Instances die folgenden Eigenschaften aufweisen:

- Sie unterstützen einen Tomcat-Anwendungsserver mit einem Apache-Frontend.
- Tomcat ist so konfiguriert, dass Anwendungen ein `DataSource` JDBC-Objekt verwenden können, um eine Verbindung zu einer separaten MySQL-Instanz herzustellen, die als Backend-Datenspeicher dient.

Das Rezeptbuch für dieses Projekt umfasst mehrere wichtige Komponenten:

- Die Attributdatei enthält Konfigurationseinstellungen, die von den verschiedenen Rezepten verwendet werden.
- Einrichtungsrezepte werden dem Setup-Lebenszyklusereignis des Layers zugewiesen. Sie werden ausgeführt, nachdem eine Instanz gestartet wurde, und führen Aufgaben wie das Installieren von Paketen und das Erstellen von Konfigurationsdateien aus.
- Konfigurationsrezepte werden dem Configure-Lebenszyklusereignis des Layers zugewiesen. Sie werden ausgeführt, nachdem sich die Konfiguration des Stacks geändert hat — in erster Linie, wenn Instances online gehen oder offline gehen — und übernehmen alle erforderlichen Konfigurationsänderungen.
- Bereitstellungsrezepte werden dem Deploy-Lebenszyklusereignis des Layers zugewiesen. Sie werden nach den Einrichtungsrezepten ausgeführt und wenn Sie eine Anwendung manuell bereitstellen, um den Code zu installieren und zugehörige Dateien auf den Instances eines Layers zu installieren, und verarbeiten die dazugehörigen Aufgaben, wie z. B. das Neustarten von Services.

Im letzten Abschnitt wird beschriebenErstellen eines Stacks und Ausführen einer Anwendung, wie Sie einen Stack erstellen, der eine benutzerdefinierte Ebene enthält, die auf dem Tomcat-Kochbuch basiert, und wie Sie eine einfache JSP-Anwendung bereitstellen und ausführen, die Daten aus einer MySQL-Datenbank anzeigt, die auf einer Instanz läuft, die zu einer separaten MySQL-Schicht gehört.

> **Note**
>
> Die Rezepte für das Tomcat-Kochbuch hängen von einigen in Stacks integrierten Rezepten ab. AWS OpsWorks Um den Ursprung der einzelnen Rezepte deutlich zu machen, werden in diesem Thema Rezepte mithilfe der Chef-Konvention Rezeptbuchname::Rezeptname bezeichnet.

Themen

- Attributdatei
- Einrichtungsrezepte
- Konfigurationsrezepte
- Bereitstellungsrezepte

- Erstellen eines Stacks und Ausführen einer Anwendung

Attributdatei

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Bevor wir Rezepte betrachten, ist es sinnvoll, zuerst die Attributdatei des Tomcat-Rezeptbuchs anzusehen, die verschiedene Konfigurationseinstellungen enthält, die die Rezepte verwenden. Attribute sind nicht erforderlich. Sie können diese Werte einfach in Ihren Rezepten oder Vorlagen fest programmieren. Wenn Sie jedoch Konfigurationseinstellungen mithilfe von Attributen definieren, können Sie die AWS OpsWorks Stacks-Konsole oder die API verwenden, um die Werte zu ändern, indem Sie benutzerdefinierte JSON-Attribute definieren. Dies ist einfacher und flexibler, als den Rezept- oder Vorlagencode jedes Mal neu zu schreiben, wenn Sie eine Einstellung ändern möchten. Dieser Ansatz ermöglicht es Ihnen beispielsweise, dasselbe Rezeptbuch für mehrere Stacks zu verwenden, aber den Tomcat-Server für jeden Stack unterschiedlich zu konfigurieren. Weitere Informationen zu Attributen und wie diese überschrieben werden können, finden Sie unter Überschreiben der Attribute.

Das folgende Beispiel zeigt die komplette Attributdatei, `default.rb`, die sich im Verzeichnis `attributes` des Tomcat-Rezeptbuchs befindet.

```
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
```

Die Einstellungen selbst werden später im entsprechenden Abschnitt besprochen. Die folgenden Hinweise gelten allgemein:

- Alle Knotendefinitionen weisen den `default`-Typ auf. Sie können also mit benutzerdefinierten JSON-Attributen überschrieben werden.
- Die Datei verwendet eine `case`-Anweisung, um einige Attributwerte basierend auf dem Betriebssystem der Instance bedingt festzulegen.

  Der `platform`-Knoten wird vom Ohai Tool von Chef generiert und stellt das Betriebssystem der Instance dar.

Einrichtungsrezepte

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Einrichtungsrezepte werden dem Setup-Lebenszyklusereignis des Layers zugeordnet und nach dem Start einer Instance ausgeführt. Sie führen Aufgaben wie das Installieren von Paketen, das Erstellen von Konfigurationsdateien und das Starten von Services durch. Nachdem die Ausführung der Setup-Rezepte abgeschlossen ist, führt AWS OpsWorks Stacks die Deploy-Rezepte aus, um alle Apps auf der neuen Instanz bereitzustellen.

Themen

- tomcat::setup
- tomcat::install
- tomcat::service
- tomcat::container_config
- tomcat::apache_tomcat_bind

tomcat::setup

Das `tomcat::setup`-Rezept ist dafür konzipiert, dem Setup-Lebenszyklusereignis eines Layer zugewiesen zu werden.

```
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'
```

Das `tomcat::setup`-Rezept ist weitestgehend ein Metarezept. Es enthält eine Reihe von abhängigen Rezepten, die die meisten Details der Installation und Konfiguration von Tomcat und zugehörigen Paketen verarbeiten. Der erste Teil von `tomcat::setup` führt die folgenden Rezepte aus, die zu einem späteren Zeitpunkt besprochen werden:

- Das tomcat::install-Rezept installiert das Tomcat-Serverpaket.
- Das tomcat::service-Rezept richtet den Tomcat-Service ein.

Der mittlere Teil von `tomcat::setup` ermöglicht und startet den Tomcat-Service:

- Die service-Ressource von Chef aktiviert den Tomcat-Service beim Start.
- Die Chef-Bash-Ressource führt ein Bash-Skript aus, um den autofs-Daemon zu starten, der für Amazon EBS-gestützte Instances erforderlich ist. Die Ressource weist dann die `service`-Ressource an, den Tomcat-Service neu zu starten.

  Weitere Informationen finden Sie unter: autofs (für Amazon Linux) oder Autofs (für Ubuntu).

Der letzte Teil von `tomcat::setup` erstellt Konfigurationsdateien und installiert und konfiguriert den Apache-Frontend-Server:

- Das tomcat::container_config-Rezept erstellt Konfigurationsdateien.
- Das `apache2` Rezept (das eine Abkürzung für`apache2::default`) ist ein in AWS OpsWorks Stacks integriertes Rezept, das einen Apache-Server installiert und konfiguriert.
- Das tomcat::apache_tomcat_bind-Rezept konfiguriert den Apache-Server so, dass er als Frontend für den Tomcat-Server dient.

> **Note**
>
> Sie können oft Zeit und Mühen sparen, indem Sie integrierte Rezepte für die Durchführung einiger der erforderlichen Aufgaben nutzen. Dieses Rezept verwendet das integrierte `apache2::default`-Rezept zum Installieren von Apache anstelle einer Implementierung von Anfang an. Ein weiteres Beispiel für die Verwendung integrierter Rezepte finden Sie unter Bereitstellungsrezepte.

Die folgenden Abschnitte beschreiben die Einrichtungsrezepte des Tomcat-Rezeptbuch im Detail. Weitere Informationen zu den `apache2`-Rezepten finden Sie unter opsworks-cookbooks/apache2.

tomcat::install

Das `tomcat::install` -Rezept installiert den Tomcat-Server, OpenJDK und eine Java-Konnektorbibliothek, die die Verbindung zum MySQL-Server verarbeitet.

```
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'
```

Das Rezept führt die folgenden Aufgaben aus:

1. Es erstellt eine Liste der Pakete, die installiert werden, abhängig vom Betriebssystem der Instance.
2. Es installiert jedes Paket auf der Liste.

   Die Chef-Paketressource verwendet den entsprechenden Anbieter — `yum` für Amazon Linux und `apt-get` für Ubuntu —, um die Installation durchzuführen. Die Paketanbieter installieren OpenJDK als Tomcat-Abhängigkeit, die MySQL-Konnektorbibliothek muss jedoch explizit installiert werden.
3. Es verwendet eine link-Ressource von Chef zum Erstellen eines symbolischen Links (symlink) im Verzeichnis "lib" des Tomcat-Servers zur MySQL-Konnektorbibliothek im JDK.

   Unter Verwendung der standardmäßigen Attributwerte lautet das Tomcat-lib-Verzeichnis `/usr/share/tomcat6/lib` und die MySQL-Konnektorbibliothek (`mysql-connector-java.jar`) befindet sich unter `/usr/share/java/`.

Das Rezept `tomcat::remove_root_webapp` entfernt die ROOT-Webanwendung (standardmäßig `/var/lib/tomcat6/webapps/ROOT`), um einige Sicherheitsprobleme zu vermeiden.

```
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
```

Die `only_if`-Anweisung stellt sicher, dass das Rezept die Datei nur dann entfernt, wenn sie vorhanden ist.

> **Note**
>
> Die Tomcat-Version wird von dem `['tomcat']['base_version']`-Attribut spezifiziert, das in der Attributdatei auf 6 festgelegt ist. Zur Installation von Tomcat 7 können Sie benutzerdefinierte JSON-Attribute verwenden, um das Attribut zu überschreiben. Bearbeiten

Sie Ihre Stack-Einstellungen und geben Sie im Feld Custom Chef JSON folgende JSON-Objekte ein oder fügen Sie ein bestehendes benutzerdefiniertes JSON-Objekt hinzu:

```
{
  'tomcat' : {
    'base_version' : 7
  }
}
```

Das benutzerdefinierte JSON-Attribut überschreibt das Standardattribut und legt die Tomcat-Version auf 7 fest. Weitere Informationen über das Überschreiben von Attributen finden Sie unter Überschreiben der Attribute.

tomcat::service

Das `tomcat::service`-Rezept erstellt die Tomcat-Servicedefinition.

```
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
```

Das Rezept verwendet die service-Ressource von Chef, um den Tomcat-Servicenamen (standardmäßig "tomcat6") anzugeben, und das `supports`-Attribut, um zu definieren, wie Chef die Neustart-, Neulade- und Statusbefehle auf den verschiedenen Betriebssystemen verwaltet.

- `true` gibt an, dass Chef das Init-Skript oder einen anderen Serviceanbieter zum Ausführen des Befehls verwenden kann.
- `false` gibt an, dass Chef versuchen muss, den Befehl anderweitig auszuführen.

Beachten Sie, dass `action` auf `:nothing` festgelegt ist. Für jedes Lebenszyklusereignis initiiert AWS OpsWorks Stacks einen Chef-Lauf, um die entsprechenden Rezepte auszuführen. Das Tomcat-Rezeptbuch folgt einem allgemeinen Muster, das festlegt, dass ein Rezept die Servicedefinition erstellt, den aber Service nicht neu startet. Andere Rezepte in der Chef-Ausführung verarbeiten den Neustart, normalerweise mit einem `notifies`-Befehl in den `template`-Ressourcen, die zum Erstellen von Konfigurationsdateien verwendet werden. Benachrichtigungen sind eine komfortable Möglichkeit, einen Service neu zu starten, denn sie tun das nur, wenn die Konfiguration geändert wurde. Wenn eine Chef-Ausführung mehrere Neustartbenachrichtigungen für einen Service aufweist, startet Chef den Service zudem höchstens einmal neu. So werden Probleme vermieden, die auftreten können, wenn versucht wird, einen Service neu zu starten, der nicht voll betriebsbereit. Dies ist eine häufige Quelle von Fehlern bei Tomcat.

Der Tomcat-Service muss für jede Chef-Ausführung, die Neustartbenachrichtigungen verwendet, definiert werden. `tomcat::service` ist daher in mehreren Rezepten enthalten, um sicherzustellen, dass der Service für jede Chef-Ausführung definiert ist. Es entstehen keine Nachteile, wenn eine Chef-Ausführung mehrere Instances von `tomcat::service` umfasst, da Chef sicherstellt, dass ein Rezept nur einmal pro Ausführung ausgeführt wird, unabhängig davon, wie oft es enthalten ist.

tomcat::container_config

Das `tomcat::container_config`-Rezept erstellt Konfigurationsdateien von Rezeptbuch-Vorlagendateien.

```
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
```

Das Rezept ruft zuerst `tomcat::service` auf, das den Service bei Bedarf definiert. Der Großteil des Rezepts besteht aus zwei template-Ressourcen, von denen jede eine Konfigurationsdatei von einer der Vorlagendateien des Rezeptbuchs erstellt, die Dateieigenschaften festlegt und Chef anweist, den Service neu zu starten.

Tomcat-Umgebungskonfigurationsdatei

Die erste `template`-Ressource verwendet die Vorlagendatei `tomcat_env_config.erb` zum Erstellen einer Tomcat-Umgebungskonfigurationsdatei, die zum Festlegen von Umgebungsvariablen wie `JAVA_HOME` verwendet wird. Der Standardname ist das Argument der `template`-Ressource. `tomcat::container_config` verwendet ein `path`-Attribut, um den Standardwert zu überschreiben und der Konfigurationsdatei den Namen `/etc/sysconfig/tomcat6` (Amazon Linux) oder `/etc/default/tomcat6` (Ubuntu) zu geben. Die `template`-Ressource gibt zudem den Eigentümer, die Gruppe und die Moduseinstellungen der Datei an und weist Chef an, keine Sicherungsdateien zu erstellen.

Wenn Sie sich den Quellcode ansehen, gibt es tatsächlich drei Versionen von `tomcat_env_config.erb`, in jeweils unterschiedlichen Unterverzeichnissen des `templates`-Verzeichnisses. Die Verzeichnisse `ubuntu` und `amazon` enthalten die Vorlagen für die jeweiligen Betriebssysteme. Der Ordner `default` enthält eine Dummy-Vorlage mit einer einzigen Kommentarzeile, die nur verwendet wird, wenn Sie versuchen, dieses Rezept für eine Instance mit einem nicht unterstützten Betriebssystem auszuführen. Das Rezept `tomcat::container_config` muss nicht angeben, welche Datei `tomcat_env_config.erb` zu verwenden ist. Chef wählt automatisch das entsprechende Verzeichnis für das Betriebssystem der Instance aus, basierend auf den unter File Specificity beschriebenen Regeln.

Die `tomcat_env_config.erb`-Dateien für dieses Beispiel bestehen größtenteils aus Kommentaren. Um zusätzliche Umgebungsvariablen festzulegen, heben Sie die Auskommentierung der entsprechenden Zeilen auf und stellen Sie Ihre bevorzugten Werte bereit.

> **Note**
>
> Jede Konfigurationseinstellung, die sich ändern könnte, sollte als Attribut definiert und nicht in der Vorlage fest programmiert werden. Auf diese Weise müssen Sie nicht die

Vorlage überschreiben, um eine Einstellung zu ändern. Sie können einfach das Attribut überschreiben.

Die Amazon Linux-Vorlage legt nur eine Umgebungsvariable fest, wie im folgenden Auszug gezeigt.

```
...
# 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 kann verwendet werden, um Java-Optionen anzugeben, wie z. B. den Bibliothekspfad. Mithilfe der standardmäßigen Attributwerte legt die Vorlage keine Java-Optionen für Amazon Linux fest. Sie können Ihre eigenen Java-Optionen festlegen, indem Sie z. B. das `['tomcat']['java_opts']`-Attribut mithilfe benutzerdefinierter JSON-Attribute überschreiben. Ein Beispiel finden Sie unter Erstellen eines Stacks.

Die Ubuntu-Vorlage legt verschiedene Umgebungsvariablen fest, wie im folgenden Auszug aus der Vorlage gezeigt.

```
# 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 -%>
```

Mithilfe von standardmäßigen Attributwerten legt die Vorlage die Ubuntu-Umgebungsvariablen wie folgt fest:

- `TOMCAT6_USER` und `TOMCAT6_GROUP`, die den Tomcat-Benutzer und die Tomcat-Gruppe darstellen, sind beide auf `tomcat6` festgelegt.

  Wenn Sie ['tomcat']['base_version'] auf `tomcat7` festlegen, werden die Variablennamen zu `TOMCAT7_USER` und `TOMCAT7_GROUP` aufgelöst und beide sind auf `tomcat7` festgelegt.
- `JAVA_OPTS` ist festgelegt auf `-Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC`.
  - Wenn Sie `-Djava.awt.headless` auf `true` festlegen, wird der Grafik-Engine mitgeteilt, dass die Instance keinen Monitor und keine Konsole hat, wodurch fehlerhaftes Verhalten bestimmter grafischer Anwendungen behoben wird.
  - `-Xmx128m` stellt sicher, dass die JVM über ausreichend Arbeitsspeicherressourcen verfügt, 128 MB für dieses Beispiel.
  - `-XX:+UseConcMarkSweepGC` legt eine gleichzeitige Mark-Sweep-Speicherbereinigung fest, was dazu beiträgt, durch Speicherbereinigung verursachte Pausen einzuschränken.

  Weitere Informationen finden Sie unter Concurrent Mark Sweep Collector Enhancements.
- Wenn die Tomcat-Version niedriger ist als 7, löscht die Vorlage die Festlegung von `LC_ALL`, wodurch ein Ubuntu-Problem gelöst wird.

> **Note**
>
> Mit den Standardattributen werden einige dieser Umgebungsvariablen einfach auf ihre Standardwerte gesetzt. Das explizite Festlegen von Umgebungsvariablen auf Attribute bedeutet jedoch, dass Sie benutzerdefinierte JSON-Attribute definieren können, um die Standardattribute zu überschreiben und benutzerdefinierte Werte bereitzustellen. Weitere Informationen über das Überschreiben von Attributen finden Sie unter Überschreiben der Attribute.

Vollständige Vorlagendateien finden Sie im Quellcode.

Konfigurationsdatei "Server.xml"

Die zweite `template`-Ressource verwendet `server.xml.erb` zum Erstellen der `system.xml` Konfigurationsdatei, die den servlet/JSP-Container konfiguriert. `server.xml.erb` enthält keine betriebssystemspezifischen Einstellungen, weshalb sie sich im `template`-Verzeichnis, im `default`-Unterverzeichnis befindet.

Die Vorlage verwendet Standardeinstellungen, kann jedoch eine Datei `system.xml` für Tomcat 6 oder für Tomcat 7 erstellen. Der folgende Code aus dem Serverabschnitt der Vorlage konfiguriert beispielsweise die entsprechenden Listener für die angegebene Version.

```
<% 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 -%>
```

Die Vorlage verwendet Attribute anstelle von fest programmierten Einstellungen, damit Sie die Einstellungen einfach ändern können, indem Sie benutzerdefinierte JSON-Attribute definieren. Beispielsweise:

```
<Connector port="<%= node['tomcat']['port'] %>" protocol="HTTP/1.1"
           connectionTimeout="20000"
           URIEncoding="<%= node['tomcat']['uri_encoding'] %>"
```

```
            redirectPort="<%= node['tomcat']['secure_port'] %>" />
```

Weitere Informationen finden Sie im Quellcode.

tomcat::apache_tomcat_bind

Das `tomcat::apache_tomcat_bind`-Rezept ermöglicht dem Apache-Server, als Tomcat-Frontend zu agieren, eingehende Anforderungen zu erhalten und sie an Tomcat weiterzuleiten sowie die Antworten an den Client zurückzugeben. Dieses Beispiel verwendet mod_proxy als Apache-Proxy/Gateway.

```
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
```

Um `mod_proxy` zu aktivieren, müssen Sie das `proxy`-Modul und ein protokollbasiertes Modul aktivieren. Es gibt zwei Möglichkeiten für das Protokollmodul:

- HTTP: `proxy_http`
- Apache JServ Protocol (AJP): `proxy_ajp`

  AJP ist ein internes Tomcat-Protokoll.

Beide execute-Ressourcen des Rezepts führen den `a2enmod`-Befehl aus, der das angegebene Modul aktiviert, indem die erforderlichen symbolischen Links (symlinks) erstellt werden:

- Die erste `execute`-Ressource aktiviert das `proxy`-Modul.
- Die zweite `execute`-Ressource aktiviert das Protokollmodul, das standardmäßig auf `proxy_http` festgelegt ist.

  Wenn Sie lieber AJP verwenden, können Sie ein benutzerdefiniertes JSON-Objekt definieren, um das `apache_tomcat_bind_mod`-Attribut zu überschreiben und es auf `proxy_ajp` festzulegen.

Das `apache2::service` Rezept ist ein in AWS OpsWorks Stacks integriertes Rezept, das den Apache-Dienst definiert. Weitere Informationen finden Sie im Rezept im AWS OpsWorks Stacks-Repository GitHub .

Die `template`-Ressource verwendet `apache_tomcat_bind.conf.erb` zum Erstellen einer Konfigurationsdatei, die standardmäßig `tomcat_bind.conf` benannt wird. Die Datei wird im Verzeichnis `['apache']['dir']/.conf.d` abgelegt. Das `['apache']['dir']`-Attribut ist in der integrierten `apache2`-Attributdatei definiert und standardmäßig auf `/etc/httpd` (Amazon Linux) bzw. `/etc/apache2` (Ubuntu) festgelegt. Wenn die `template`-Ressource die Konfigurationsdatei erstellt oder ändert, plant der `notifies`-Befehl einen Neustart des Apache-Services.

```
<% 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 -%>
```

Die Vorlage verwendet die ProxyPassReverseDirektiven ProxyPassund, um den Port zu konfigurieren, der für die Weiterleitung des Datenverkehrs zwischen Apache und Tomcat verwendet wird. Da sich beide Server auf derselben Instance befinden, können sie eine "localhost"-URL verwenden und sind beide standardmäßig auf `http://localhost:8080` festgelegt.

Konfigurationsrezepte

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Konfigurationsrezepte werden dem Configure-Lebenszyklusereignis des Layers zugewiesen, das auf allen Instances des Stacks stattfindet, wenn eine Instance in den Online-Status wechselt oder diesen verlässt. Sie verwenden Konfigurationsrezepte, um die Konfiguration einer Instance so anzupassen, dass in entsprechender Weise auf die Änderung reagiert wird. Wenn Sie ein Konfigurationsrezept implementieren, sollten Sie bedenken, dass sich eine Stack-Konfigurationsänderung möglicherweise auf Instances auswirkt, die nichts mit diesem Layer zu tun haben. Das Rezept muss entsprechend reagieren können, was in einigen Fällen auch bedeuten kann, dass nichts durchgeführt wird.

tomcat::configure

Das `tomcat::configure`-Rezept ist für das Configure-Lebenszyklusereignis eines Layers bestimmt.

```
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'
```

Das `tomcat::configure`-Rezept ist grundsätzlich ein Metarezept, das zwei abhängige Rezepte ausführt.

1. Das `tomcat::context`-Rezept erstellt eine Webanwendungs-Kontextkonfigurationsdatei.

Diese Datei konfiguriert die JDBC-Ressourcen, die Anwendungen verwenden, um mit der MySQL-Instance zu kommunizieren, wie im nächsten Abschnitt beschrieben. Wenn Sie dieses Rezept als Reaktion auf ein Konfigurationsereignis ausführen, kann der Layer die Webanwendungs-Kontextkonfigurationsdatei aktualisieren, wenn sich der Datenbank-Layer geändert hat.

2. Das `tomcat::container_config`-Einrichtungsrezept wird nochmals ausgeführt, um Änderungen in der Container-Konfiguration zu erfassen.

Das `include` für `tomcat::container_config` wird für dieses Beispiel auskommentiert. Wenn Sie ein benutzerdefiniertes JSON-Objekt zum Ändern von Tomcat-Einstellungen verwenden möchten, können Sie den Kommentar entfernen. Ein Configure-Lebenszyklusereignis führt sogar `tomcat::container_config` aus, wodurch die zu Tomcat gehörenden Konfigurationsdateien aktualisiert werden, wie in tomcat::container_config beschrieben, und startet den Tomcat-Service neu.

tomcat::context

Das Tomcat-Kochbuch ermöglicht es Anwendungen, mithilfe eines DataSourceJ2EE-Objekts auf einen MySQL-Datenbankserver zuzugreifen, der auf einer separaten Instanz ausgeführt werden kann. Mit Tomcat können Sie die Verbindung aktivieren, indem Sie eine Webanwendungs-Kontextkonfigurationsdatei für jede Anwendung erstellen und installieren. Diese Datei definiert die Beziehung zwischen der Anwendung und der JDBC-Ressource, die die Anwendung für die Kommunikation mit der Datenbank verwendet. Weitere Informationen finden Sie unter The Context Container.

Der Hauptzweck des `tomcat::context`-Rezepts ist die Erstellung dieser Konfigurationsdatei.

```
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
```

Zusätzlich zu den Tomcat-Kochbuchattributen verwendet dieses Rezept die Stack-Konfiguration und die Bereitstellungsattribute, die Stacks mit dem Configure-Ereignis AWS OpsWorks installiert. Der AWS OpsWorks Stacks-Dienst fügt dem Knotenobjekt jeder Instanz Attribute hinzu, die die Informationen enthalten, die Rezepte normalerweise mithilfe von Datenbeuteln oder Suchen abrufen würden, und installiert die Attribute auf jeder Instanz. Die Attribute enthalten detaillierte Informationen über die Stack-Konfiguration, bereitgestellte Apps und benutzerdefinierte Daten, die ein Benutzer einbeziehen möchte. Rezepte können Daten von Attributen der Stack-Konfiguration und -Bereitstellung mithilfe von standardmäßiger Chef-Knotensyntax abrufen. Weitere Informationen finden Sie unter Attribute für die Stack-Konfiguration und -Bereitstellung. Mit Chef 11.10-Stacks können Sie auch die Chef-Suche verwenden, um Daten der Stack-Konfiguration und -Bereitstellung abzurufen. Weitere Informationen finden Sie unter Verwenden der Chef-Suchfunktion.

`deploy`attributes bezieht sich auf den `[:deploy]` Namespace, der bereitstellungsbezogene Attribute enthält, die über die Konsole oder API definiert oder vom Stacks-Dienst generiert werden. AWS OpsWorks Das `deploy`-Attribut enthält ein Attribut für jede bereitgestellte Anwendung, wobei die Kurzbezeichnung der Anwendung verwendet wird. Jedes Anwendungsattribut enthält eine Reihe von Attributen, die die Anwendung charakterisieren, wie z. B. das Dokumenten-Stammverzeichnis (`[:deploy][:`*`appname`*`][:document_root]`).

Das `context`-Rezept stellt zuerst sicher, dass der Service für diese Chef-Ausführung definiert ist, indem tomcat::service aufgerufen wird. Anschließend definiert es eine `context_name`-Variable, die den Namen der Konfigurationsdatei darstellt, ohne die `.xml`-Erweiterung. Wenn Sie das standardmäßige Dokumenten-Stammverzeichnis verwenden, wird `context_name` auf den Kurznamen der Anwendung festgelegt. Andernfalls wird es auf das angegebene Dokumenten-Stammverzeichnis festgelegt. In dem in Erstellen eines Stacks und Ausführen einer Anwendung erläuterten Beispiel wird das Dokumenten-Stammverzeichnis auf `"ROOT"` festgelegt, sodass der Kontext "ROOT" ist und die Konfigurationsdatei `ROOT.xml` benannt wird.

Der Großteil des Rezepts geht die Liste der bereitgestellten Anwendungen durch und verwendet für jede Anwendung die Vorlage `webapp_context.xml.erb` zur Erstellung einer Kontextkonfigurationsdatei. Das Beispiel stellt nur eine Anwendung bereit, die Definition des `deploy`-Attributs erfordert aber dennoch, dass Sie es als eine Liste von Anwendungen behandeln.

Die Vorlage `webapp_context.xml.erb` ist nicht betriebssystemspezifisch. Sie befindet sich also im Unterverzeichnis `templates` des Verzeichnisses `default`.

Das Rezept erstellt die Konfigurationsdatei wie folgt:

- Unter Verwendung von Standardattributwerten wird der Name der Konfigurationsdatei auf *`context_name`*`.xml` festgelegt und im Verzeichnis `/etc/tomcat6/Catalina/localhost/` installiert.

  Der `['datasources']`-Knoten aus den Stack-Konfigurationsattributen enthält ein oder mehrere Attribute, die jeweils einen Kontextnamen der JDBC-Datenressource zuordnen, die die zugeordnete Anwendung für die Kommunikation mit der Datenbank verwendet. Der Knoten und sein Inhalt werden beim Erstellen des Stacks mit benutzerdefinierter JSON definiert, wie später in Erstellen eines Stacks und Ausführen einer Anwendung erläutert. Das Beispiel hat ein einzelnes Attribut, das den Kontextnamen "ROOT" einer JDBC-Ressource mit dem Namen "jdbc/mydb" zuordnet.
- Mithilfe von Standardattributwerten werden sowohl der Benutzer als auch die Gruppe der Datei auf die vom Tomcat-Paket definierten Werte festgelegt: `tomcat` (Amazon Linux) oder `tomcat6` (Ubuntu).
- Die `template`-Ressource erstellt die Konfigurationsdatei nur dann, wenn der `['datasources']`-Knoten vorhanden ist und ein `context_name`-Attribut enthält.
- Die Ressource `template` definiert die beiden Variablen `resource_name` und `webapp_name`.

  `resource_name` wird auf den Ressourcennamen festgelegt, der `context_name` zugeordnet ist, und `webapp_name` auf den Kurznamen der Anwendung festgelegt.
- Die template-Ressource startet den Tomcat-Service neu, um die Änderungen zu laden und zu aktivieren.

Die Vorlage `webapp_context.xml.erb` besteht aus einem `Context`-Element, das ein `Resource`-Element mit einem eigenen Satz an Attributen enthält.

Die `Resource` Attribute kennzeichnen die Kontextkonfiguration:

- name — Der Name der JDBC-Ressource, der auf den in definierten `resource_name` Wert gesetzt ist. `tomcat::context`

  Für das Beispiel wird der Ressourcenname auf "jdbc/mydb" festgelegt.
- auth und type — Dies sind Standardeinstellungen für JDBC-Verbindungen. `DataSource`

- MaxActive, MaxIdle und MaxWait — Die maximale Anzahl von aktiven und inaktiven Verbindungen sowie die maximale Wartezeit, bis eine Verbindung zurückgegeben wird.
- username und password — Der Benutzername und das Root-Passwort der Datenbank, die aus den Attributen abgerufen werden. `deploy`
- driverClassName— Der Klassenname des JDBC-Treibers, der auf den MySQL-Treiber gesetzt ist.
- url — Die Verbindungs-URL.

  Das Präfix hängt von der Datenbank ab. Es sollte folgendermaßen festgelegt werden: `jdbc:mysql` für MySQL, `jdbc:postgresql` für Postgres und `jdbc:sqlserver` für SQL Server. Im Beispiel wird die URL auf `jdbc:mysql://`*`host_IP_Address`*`:3306:simplejsp` festgelegt, wobei *`simplejsp`* der Kurzname der App ist.
- factory — Die `DataSource` Factory, die für MySQL-Datenbanken erforderlich ist.

Weitere Informationen zu dieser Konfigurationsdatei finden Sie im DataSources Thema Using des Tomcat-Wikis.

Bereitstellungsrezepte

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Bereitstellungsrezepte werden dem Deploy-Lebenszyklusereignis des Layers zugewiesen. Es tritt normalerweise auf allen Instanzen des Stacks auf, wann immer Sie eine App bereitstellen. Sie können das Ereignis jedoch optional auf nur bestimmte Instanzen beschränken. AWS OpsWorks Stacks führt die Deploy-Rezepte auch auf neuen Instanzen aus, nachdem die Setup-Rezepte abgeschlossen sind. Der Hauptzweck von Bereitstellungsrezepten besteht in der Bereitstellung von Code und zugehörigen Dateien aus einem Repository in den Instances des Anwendungsserver-Layers. Bereitstellungsrezepte werden jedoch auch oft auf anderen Layern ausgeführt. Auf diese Weise können Instances dieser Layer z. B. ihre Konfiguration aktualisieren, um die neu bereitgestellte Anwendung einzubinden. Wenn Sie ein Bereitstellungsrezept implementieren, beachten Sie, dass

ein Deploy-Ereignis nicht notwendigerweise bedeutet, dass der Instance Anwendungen bereitgestellt werden. Es könnte auch einfach nur eine Benachrichtigung sein, dass Anwendungen in anderen Instances im Stack bereitgestellt werden, um zu ermöglichen, dass die Instance notwendige Updates durchführt. Das Rezept muss entsprechend reagieren können, was auch bedeuten kann, dass nichts durchgeführt wird.

AWS OpsWorks Stacks stellt Apps der Standard-App-Typen automatisch auf den entsprechenden integrierten Anwendungsserverschichten bereit. Zur Bereitstellung von Anwendungen in einem benutzerdefinierten Layer müssen Sie benutzerdefinierte Bereitstellungsrezepte implementieren, die die Dateien der Anwendung von einem Repository in den entsprechenden Speicherort in der Instance herunterladen. Sie können jedoch häufig die Menge des Codes begrenzen, den Sie schreiben müssen, indem Sie das integrierte Bereitstellungsrezeptbuch verwenden, um verschiedene Aspekte der Bereitstellung zu verarbeiten. Wenn Sie beispielsweise Ihre Dateien in einem der unterstützten Repositorys speichern, kann das integrierte Rezeptbuch die Details des Herunterladens von Dateien aus dem Repository in die Instances des Layers verarbeiten.

Das `tomcat::deploy`-Rezept ist dafür konzipiert, dem Deploy-Lebenszyklusereignis zugewiesen zu werden.

```
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
...
```

Das `tomcat::deploy`-Rezept verwendet das integrierte Bereitstellungsrezeptbuch für Aspekte der Bereitstellung, die nicht anwendungsspezifisch sind. Das `deploy`-Rezept (die Kurzbezeichnung für das integrierte `deploy::default`-Rezept) ist ein integriertes Rezept, das die Details der Einrichtung der Benutzer, Gruppen usw. verarbeitet, basierend auf Daten aus den `deploy`-Attributen.

Das Rezept verwendet zwei integrierte Chef-Definitionen, `opsworks_deploy_dir` und `opworks_deploy`, zum Installieren der Anwendung.

Die `opsworks_deploy_dir`-Definition richtet die Verzeichnisstruktur basierend auf Daten der JSON-Bereitstellung der Anwendung ein. Definitionen sind grundsätzlich eine bequeme Möglichkeit, Ressourcendefinitionen zu verpacken, und befinden sich im Verzeichnis `definitions` eines Rezeptbuchs. Rezepte können Definitionen ähnlich wie Ressourcen verwenden, aber der Definition selbst ist kein Anbieter zugeordnet, sondern nur die Ressourcen, die in der Definition enthalten sind. Sie können Variablen im Rezept definieren, die an die zugrunde liegenden Ressourcendefinitionen weitergegeben werden. Das `tomcat::deploy`-Rezept legt die Variablen `user`, `group` und `path` basierend auf Daten aus der JSON-Bereitstellung fest. Sie werden an die directory-Ressource der Definition weitergegeben, die die Verzeichnisse verwaltet.

> **Note**
>
> Die Benutzer und die Gruppe Ihrer bereitgestellten Anwendung werden von den Attributen`[:opsworks][:deploy_user][:user]` und `[:opsworks][:deploy_user][:group]` bestimmt, die in der Attributdatei des integrierten Bereitstellungsrezeptbuchs `deploy.rb` definiert werden. Der Standardwert von `[:opsworks][:deploy_user][:user]` ist `deploy`. Der Standardwert von `[:opsworks][:deploy_user][:group]` hängt vom Betriebssystem der Instance ab:
>
> - Für Ubuntu-Instances ist die Standardgruppe `www-data`.
> - Für Amazon Linux-Instances, die Mitglieder einer Rails-App Server-Ebene sind, die Nginx und Unicorn verwendet, ist die Standardgruppe. `nginx`
> - Für alle anderen Amazon Linux-Instances ist die Standardgruppe `apache`.
>
> Sie können diese Einstellungen ändern, indem Sie mithilfe einer benutzerdefinierten JSON-Datei oder einer benutzerdefinierten Attributdatei das entsprechende Attribut überschreiben. Weitere Informationen finden Sie unter Überschreiben der Attribute.

Die andere Definition, `opsworks_deploy`, verarbeitet die Details der Überprüfung des Codes der App und der zugehörigen Dateien aus dem Repository und der Bereitstellung in der Instance, basierend auf Daten aus den `deploy`-Attributen. Sie können diese Definition für jeden Anwendungstyp verwenden. Bereitstellungsdetails wie die Verzeichnisnamen werden in der Konsole oder über die API festgelegt und in den `deploy`-Attributen gespeichert. Allerdings

funktioniertopsworks_deploy nur für die vier unterstützten Repository-Typen: Git, Subversion, S3 und HTTP. Sie müssen diesen Code selbst implementieren, wenn Sie einen anderen Repository-Typ verwenden möchten.

Sie installieren die Dateien einer App im Tomcat-Verzeichnis webapps. Eine typische Methode ist es, Dateien direkt nach webapps zu kopieren. Die AWS OpsWorks Stacks-Bereitstellung ist jedoch so konzipiert, dass bis zu fünf Versionen einer App auf einer Instance beibehalten werden, sodass Sie bei Bedarf zu einer früheren Version zurückkehren können. AWS OpsWorks Stacks macht daher Folgendes:

1. Es stellt Apps in einem getrennten Verzeichnis bereit, dessen Name einen Zeitstempel enthält, wie z. B. /srv/www/my_1st_jsp/releases/20130731141527.
2. Es erstellt einen symlink mit dem Namen current, wie etwa /srv/www/my_1st_jsp/current, zu diesem eindeutigen Verzeichnis.
3. Wenn nicht bereits vorhanden, erstellt es einen symlink von dem Verzeichnis webapps zum in Schritt 2 erstellten symlink current.

Wenn Sie eine frühere Version wiederherstellen müssen, modifizieren Sie den symlink current so, dass er auf ein bestimmtes Verzeichnis mit dem entsprechenden Zeitstempel zeigt, etwa indem Sie das Linkziel /srv/www/my_1st_jsp/current abändern.

Im mittleren Bereich von tomcat::deploy wird der symlink eingerichtet.

```
 ...
 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
  ...
```

Das Rezept erstellt zunächst zwei Variablen, `current_dir` und `webapp_dir`, um jeweils die Verzeichnisse `current` und `webapp` darzustellen. Dann wird eine `link`-Ressource verwendet, um `webapp_dir` mit `current_dir` zu verknüpfen. Das AWS OpsWorks `deploy::default` Stacks-Rezept erstellt einige Stub-Verzeichnisse, die für dieses Beispiel nicht erforderlich sind, sodass sie im mittleren Teil des Auszugs entfernt werden.

Der letzte Teil von `tomcat::deploy` startet den Tomcat-Service bei Bedarf neu.

```
  ...
  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'
```

Das erste Rezept führt zuerst `tomcat::service` aus, um sicherzustellen, dass der Service für diese Chef-Ausführung definiert ist. Dann wird eine execute-Ressource verwendet, um den Service anzuweisen, neu zu starten, aber nur, wenn `['tomcat']['auto_deploy']` festgelegt ist auf `'true'`. Andernfalls überwacht Tomcat Änderungen in seinem Verzeichnis `webapps`, was einen expliziten Neustart des Tomcat-Services überflüssig macht.

> **Note**
>
> Die `execute`-Ressource führt nichts wirklich Substantielles aus. `/bin/true` ist ein Dummy-Shell-Skript, das einfach einen Erfolgscode zurückgibt. Es wird hier als eine bequeme Möglichkeit verwendet, eine Neustartbenachrichtigung zu generieren. Wie bereits erwähnt wird durch die Verwendung von Benachrichtigungen sichergestellt, dass Services nicht zu häufig neu gestartet werden.

Schließlich wird `tomcat::deploy` von `tomcat::context` ausgeführt, wodurch die Webanwendungs-Kontextkonfigurationsdatei aktualisiert wird, wenn Sie die Backend-Datenbank geändert haben.

Erstellen eines Stacks und Ausführen einer Anwendung

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Abschnitt wird erläutert, wie Sie das Tomcat-Rezeptbuch zum Implementieren einer grundlegenden Stack-Einrichtung verwenden, die eine einfache JSP-Anwendung (Java Server Pages-Anwendung) mit dem Namen "SimpleJSP" ausführt. Der Stack besteht aus einer Tomcat-basierten benutzerdefinierten Ebene mit dem Namen TomCustom und einer MySQL-Schicht. SimpleJSP wird in der MySQL-Datenbank bereitgestellt TomCustom und zeigt einige Informationen aus der MySQL-Datenbank an. Wenn Sie noch nicht mit den Grundlagen der Verwendung von AWS OpsWorks Stacks vertraut sind, sollten Sie zuerst lesen. Erste Schritte mit Chef 11 Linux-Stacks

Die Anwendung SimpleJSP

Die SimpleJSP-Anwendung zeigt die Grundlagen der Einrichtung einer Datenbankverbindung und des Abrufens von Daten aus der MySQL-Datenbank des Stacks.

```
<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 verwendet ein `DataSource`-Objekt für die Kommunikation mit der MySQL-Datenbank. Tomcat verwendet die Daten in der Webanwendungs-Kontextkonfigurationsdatei, um ein

`DataSource`-Objekt zu erstellen und zu initialisieren und mit einem logischen Namen zu verknüpfen. Es registriert dann den logischen Namen mit einer Java Naming and Directory Interface (JNDI). Um eine Instance des entsprechenden `DataSource`-Objekts zu erhalten, erstellen Sie ein `InitialContext`-Objekt und geben Sie den logischen Namen der Ressource an die `lookup`-Methode des Objekts weiter, die das entsprechende Objekt abruft. Der logische Namen des SimpleJSP-Beispiels, `java:comp/env/jdbc/mydb`, weist folgende Bestandteile auf:

- Den Stamm-Namespace, `java`, der vom restlichen Namen durch einen Doppelpunkt (:) getrennt ist
- Alle zusätzlichen Namespaces, getrennt durch Schrägstriche (/)

  Tomcat fügt dem `comp/env`-Namespace automatisch Ressourcen hinzu.
- Den Ressourcennamen, der in der Webanwendungs-Kontextkonfigurationsdatei definiert ist und mit einem Schrägstrich (/) vom Namespace getrennt ist

  Der Ressourcenname für dieses Beispiel lautet `jdbc/mydb`.

Zum Herstellen einer Verbindung mit der Datenbank führt SimpleJSP Folgendes aus:

1. Ruft die `DataSource`-Methode des `getConnection`-Objekts auf, die ein `Connection`-Objekt zurückgibt.
2. Ruft die `Connection`-Methode des `createStatement`-Objekts auf, um ein `Statement`-Objekt zu erstellen, das Sie zur Kommunikation mit der Datenbank verwenden.
3. Kommuniziert mit der Datenbank, indem die entsprechende `Statement`-Methode aufgerufen wird.

   SimpleJSP ruft `executeQuery` auf, um eine SHOW DATABASES-Abfrage auszuführen, die die Datenbanken des Servers auflistet.

Die `executeQuery`-Methode gibt ein `ResultSet`-Objekt zurück, das die Abfrageergebnisse enthält. SimpleJSP ruft den Datenbanknamen aus dem zurückgegebenen `ResultSet`-Objekts ab und verkettet sie, um eine Ausgabezeichenfolge zu erstellen. Zuletzt schließt das Beispiel die Objekte `ResultSet`, `Statement` und `Connection`. Weitere Informationen zu JSP und JDBC finden Sie unter JavaServer Pages Technology bzw. JDBC Basics.

Um SimpleJSP mit einem Stack zu verwenden, müssen Sie es in einem Repository ablegen. Sie können jedes der unterstützten Repositorys verwenden, aber um SimpleJSP mit dem Beispiel-Stack zu verwenden, der im folgenden Abschnitt besprochen wird, müssen Sie SimpleJSP in einem

öffentlichen S3-Archiv speichern. Weitere Informationen zur Verwendung der anderen Standard-Repositorys finden Sie unter Rezeptbuch-Repositorys.

Speichern von SimpleJSP in einem S3-Archiv-Repository

1. Kopieren Sie den Beispielcode in eine Datei mit dem Namen `simplejsp.jsp` und speichern Sie die Datei in einem Verzeichnis mit dem Namen `simplejsp`.
2. Erstellen Sie ein `.zip`-Archiv des Verzeichnisses `simplejsp`.
3. Erstellen Sie einen öffentlichen Amazon S3 S3-Bucket, laden `simplejsp.zip` Sie ihn in den Bucket hoch und machen Sie die Datei öffentlich.

   Eine Beschreibung der Durchführung dieser Aufgabe finden Sie unter Erste Schritte mit Amazon Simple Storage Service.

Erstellen eines Stacks

Zum Ausführen von SimpleJSP benötigen Sie einen Stack mit den folgenden Layern.

- Einen MySQL-Layer, der den Backend-MySQL-Server unterstützt
- Einen benutzerdefinierten Layer, der das Tomcat-Rezeptbuch verwendet, um Tomcat-Server-Instances zu unterstützen

So erstellen Sie den Stack

1. Klicken Sie im AWS OpsWorks Stacks-Dashboard auf Stack hinzufügen, um einen neuen Stack zu erstellen, und klicken Sie auf Erweitert **>>**, um alle Optionen anzuzeigen. Konfigurieren Sie den Stack wie folgt.

   - Name — Ein benutzerdefinierter Stackname; in diesem Beispiel wird. TomStack
   - Benutzerdefinierte Chef-Kochbücher verwenden — Stellen Sie den Schalter auf Ja, wodurch einige zusätzliche Optionen angezeigt werden.
   - Repository-Typ —Git.
   - Repository-URL —`git://github.com/amazonwebservices/opsworks-example-cookbooks.git`.
   - Custom Chef JSON — Fügen Sie den folgenden JSON hinzu:

```
{
  "tomcat": {
    "base_version": 7,
    "java_opts": "-Djava.awt.headless=true -Xmx256m"
  },
  "datasources": {
    "ROOT": "jdbc/mydb"
  }
}
```

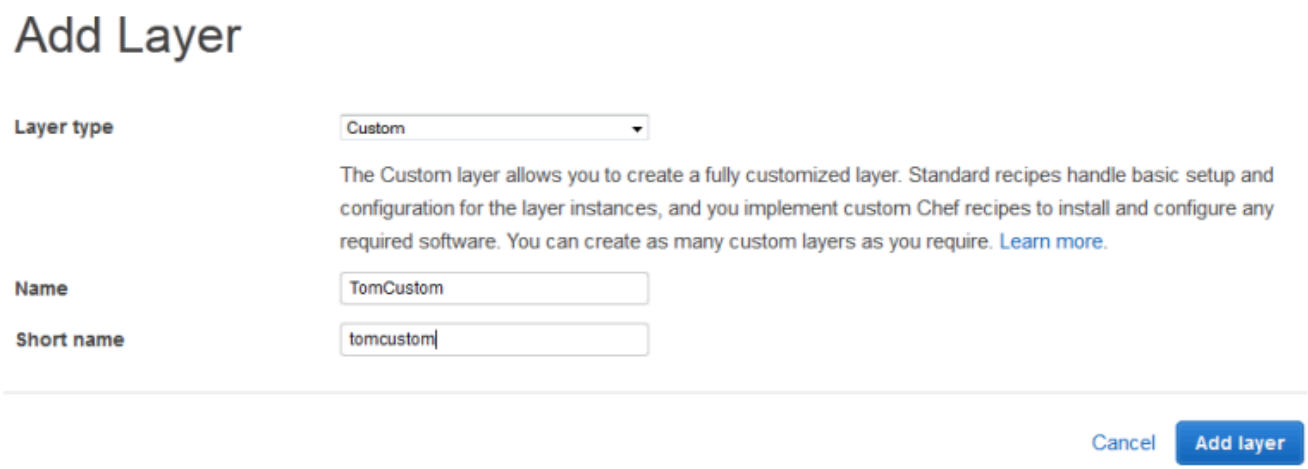
Für die restlichen Optionen können Sie die Standardwerte übernehmen.

Das benutzerdefinierte JSON-Objekt führt Folgendes durch:

- Überschreibt das `['base_version']`-Attribut des Tomcat-Rezeptbuchs, um die Tomcat-Version auf 7 festzulegen. Der Standardwert ist 6.
- Überschreibt das `['java_opts']`-Attribut des Tomcat-Rezeptbuchs, um festzulegen, dass die Instance keinen Monitor hat, und legt die maximale JVM-Heap-Größe auf 256 MB fest. Der Standardwert legt keine Optionen für Instances fest, die Amazon Linux ausführen.
- Gibt den `['datasources]`-Attributwert an, der dem Webanwendungs-Kontextnamen ("ROOT") einen JDBC-Ressourcennamen ("jdbc/mydb") zuweist, wie unter tomcat::context erläutert.

  Dieses letzte Attribut hat keinen Standardwert. Sie müssen es mit benutzerdefinierter JSON festlegen.

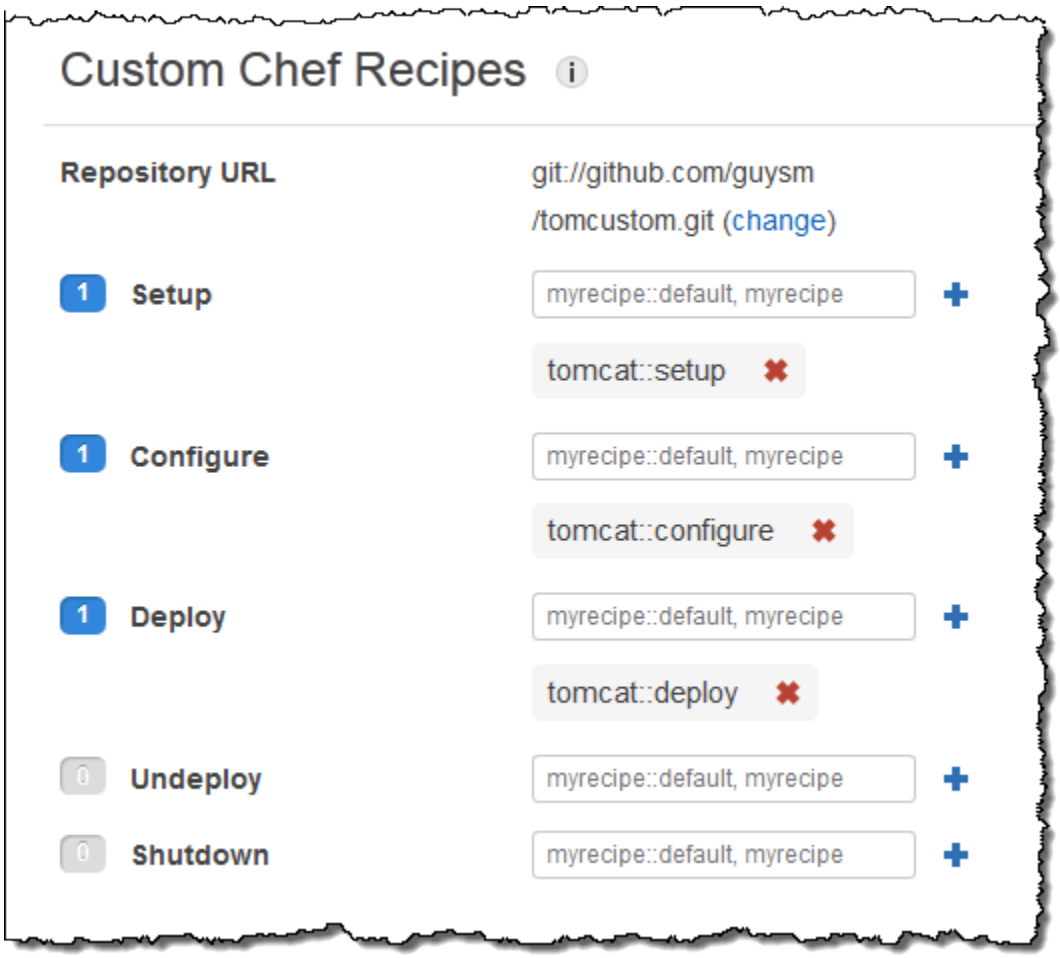
2. Klicken Sie auf Add a layer (Einen Layer hinzufügen). Wählen Sie für Layer type (Ebenentyp) die Option MySQL aus. Klicken Sie dann auf Add Layer (Ebene hinzufügen).
3. Klicken Sie im Navigationsbereich auf Instances und dann auf Add an instance (Eine Instance hinzufügen). Klicken Sie auf Add Instance (Instance hinzufügen), um die Standardeinstellungen zu übernehmen. Klicken Sie in der Zeile für die Instance auf start (starten).
4. Kehren Sie zur Seite Layers (Ebenen) zurück und klicken Sie auf + Layer (+Ebene), um einen Layer hinzuzufügen. Klicken Sie für Layer type (Ebenentyp) auf Custom (Benutzerdefiniert). Das Beispiel verwendet **`TomCustom`** bzw. **`tomcustom`** als Namen bzw. Kurznamen des Layers.



5. Klicken Sie auf der Seite Layers (Ebenen) für den entsprechenden benutzerdefinierten Layer auf Recipes (Rezepte) und dann auf Edit (Bearbeiten). Weisen Sie unter Custom Chef Recipes (Benutzerdefinierte Chef-Rezepte) den Lebenszyklusereignissen des Layers Tomcat-Rezeptbuchrezepte wie folgt zu:

   - Für Setup (Einrichtung) geben Sie tomcat::setup**`tomcat::setup`** ein und klicken Sie auf +.
   - Für Configure (Konfigurieren) geben Sie **`tomcat::configure`** ein und klicken Sie auf +.
   - Für Deploy (Bereitstellen) geben Sie **`tomcat::deploy`** ein und klicken Sie auf +. Klicken Sie dann auf Save (Speichern).

   .



6. Klicken Sie im Navigationsbereich auf Apps und dann auf Add an app (Eine App hinzufügen). Geben Sie die folgenden Optionen an und klicken Sie dann auf Add App (App hinzufügen):

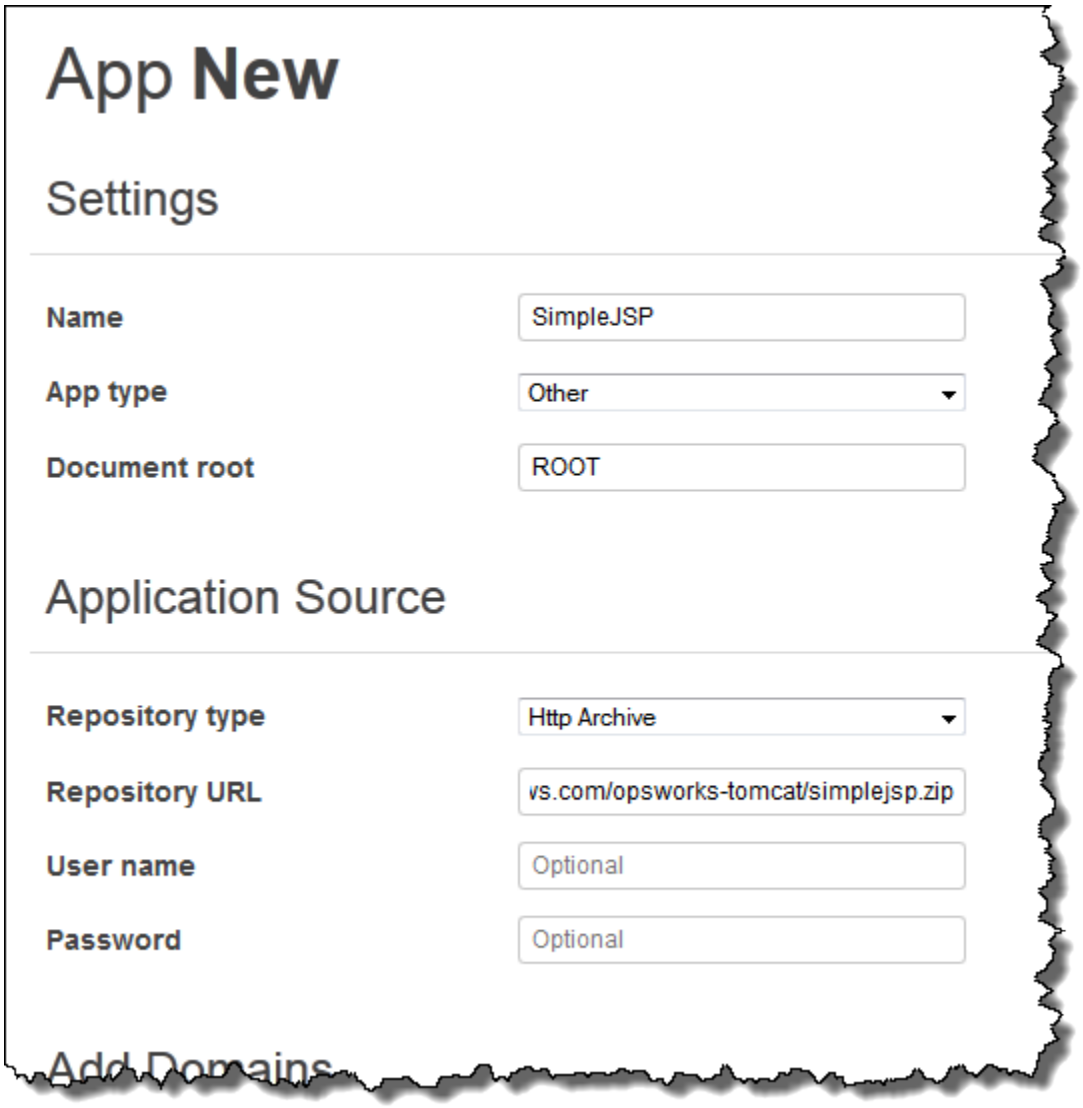
   - Name — Der Name der App; im Beispiel wird SimpleJSP verwendet und der von AWS OpsWorks Stacks generierte Kurzname lautet simplejsp.
   - App-Typ — Stellen Sie diese Option auf Andere ein.

     AWS OpsWorks Stacks stellt automatisch Standard-App-Typen auf den zugehörigen Serverinstanzen bereit. Wenn Sie App type (Typ hinzufügen) auf "Other (Andere)" festlegen, führt AWS OpsWorks Stacks einfach die Bereitstellungsrezepte aus und lässt diese die Bereitstellung verarbeiten.
   - Dokumentenstamm — Stellen Sie diese Option auf ein. **ROOT**

     Der Wert Document root (Dokumentenstamm) gibt den Kontextnamen an.
   - Repository-Typ — Stellen Sie diese Option auf S3 Archive ein.

- Repository-URL — Stellen Sie hier die Amazon S3 S3-URL der App ein, die Sie zuvor erstellt haben.

Verwenden Sie für die anderen Optionen die Standardeinstellungen.



7. Verwenden Sie die Seite „Instances“, um dem TomCustom Layer eine Instance hinzuzufügen und sie zu starten. AWS OpsWorks Stacks führt die Deploy-Rezepte nach Abschluss der Setup-Rezepte automatisch auf einer neuen Instance aus, sodass beim Starten der Instanz auch SimpleJSP bereitgestellt wird.
8. Wenn die TomCustom Instanz online ist, klicken Sie auf der Instanzenseite auf den Instanznamen, um die zugehörigen Details zu sehen. Kopieren Sie die öffentliche IP-Adresse. Anschließend erstellen Sie folgende URL: "http://*publicIP*/tc/*appname.jsp*". Für das Beispiel sieht diese URL etwa folgendermaßen aus: **`http://50.218.191.172/tc/simplejsp.jsp`**.

> **Note**
>
> Die Apache-URL, die Anfragen an Tomcat weiterleitet, ist auf das `['tomcat']['apache_tomcat_bind_path']`-Standardattribut, `/tc/`, festgelegt. Das SimpleJSP-Dokumenten-Stammverzeichnis ist auf `ROOT` festgelegt, einen speziellen Wert, der aufgelöst wird in `/`. Die URL lautet daher "... /tc/simplejsp.jsp".

9. Fügen Sie die URL aus dem vorherigen Schritt in Ihren Browser ein. Sie sollten Folgendes sehen:

```
Databases found:
information_schema
simplejsp
test
```

> **Note**
>
> Wenn Ihr Stack über eine MySQL-Instanz verfügt, erstellt AWS OpsWorks Stacks automatisch eine Datenbank für jede App, die mit dem Kurznamen der App benannt ist.

Attribute für die Stack-Konfiguration und -Bereitstellung

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn AWS OpsWorks Stacks einen Befehl auf einer Instance ausführt — zum Beispiel einen Deploy-Befehl als Reaktion auf ein Deploy-Lifecycle-Ereignis — fügt es dem Knotenobjekt der Instanz eine Reihe von Attributen hinzu, die die aktuelle Konfiguration des Stacks beschreiben. Für die Stack-Befehle Deploy-Ereignisse und Execute Recipes installiert AWS OpsWorks

Stacks Deploy-Attribute, die zusätzliche Informationen zur Bereitstellung bereitstellen. Weitere Informationen zum Knotenobjekt finden Sie unter Überschreiben der Attribute. Eine Liste der häufig verwendeten Attribute für die Stack-Konfiguration und -Bereitstellung, einschließlich qualifizierter Knotennamen, finden Sie unter Stack-Konfigurations- und Bereitstellungsattribute: Linux und Integrierte Rezeptbuchattribute.

> **Note**
>
> Für Linux-Stacks erhalten Sie eine vollständige Liste dieser Attribute, formatiert als JSON-Objekt, indem Sie den CLI-Befehl get_json des Agenten verwenden.

In den folgenden Abschnitten werden die Attribute gezeigt, die einem Konfigurations- oder Bereitstellungsereignis für einen einfachen Stack zugeordnet sind und Folgendes enthalten:

- Eine PHP-App-Server-Ebene mit zwei Instanzen
- Eine HAProxy-Schicht mit einer Instanz

Die Beispiele stammen aus einer der PHP App Server-Instanzen, php-app1. Die Attribute sind der Einfachheit halber als JSON-Objekt formatiert. Die Objektstruktur wird dem vollqualifizierten Namen der Attribute zugeordnet. So sieht das `node[:opsworks][:ruby_version]`-Attribut beispielsweise wie folgt in der JSON-Darstellung aus:

```
{
  "opsworks": {
    ...
    "ruby_version": "1.8.7",
    ...
  }
}
```

Themen

- Konfigurieren von Attributen
- Bereitstellungsattribute

Konfigurieren von Attributen

Das folgende JSON-Objekt zeigt die Attribute für ein Konfigurationsereignis, das auf jeder Instance im Stack ausgelöst wird, wenn eine Instance online oder offline geht. Die Attribute enthalten die integrierten Attribute für die Stack-Konfiguration und alle benutzerdefinierten JSON-Attribute, die für den Stack vor dem Ereignis konfiguriert wurden (in diesem Fall keine). Es wurde aus Gründen der Länge bearbeitet. Eine detaillierte Beschreibung verschiedener Attribute finden Sie unter Stack-Konfigurations- und Bereitstellungsattribute: Linux und Integrierte Rezeptbuchattribute.

```
{
  "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": {
  }
}
```

Die meisten Informationen finden Sie unter dem `opsworks`-Attribut, das häufig auch als Namespace bezeichnet wird. In der folgenden Liste werden die wichtigsten Attribute beschrieben:

- `layers`Attribute — Ein Satz von Attributen, von denen jedes die Konfiguration einer der Ebenen des Stacks beschreibt.

  Die Layer werden in diesem Beispiel über ihre Kurzbezeichnungen `php-app` und `lb` identifiziert. Weitere Informationen zu Kurzbezeichnungen für andere Layer finden Sie unter AWS OpsWorks Stacks-Ebenenreferenz.

- `instances`Attribute — Jede Ebene hat ein `instances` Element, das ein Attribut für jede der Online-Instanzen der Ebenen enthält, das mit dem Kurznamen der Instanz benannt ist.

  Die PHP App Server-Ebene besteht aus zwei Instanzen, `php-app1` und`php-app2`. Die HAProxy-Schicht hat eine Instanz,`lb1`.

  > **Note**
  >
  > Das `instances`-Element enthält nur die Instances, die online sind, wenn die entsprechenden Stack- und Bereitstellungsattribute erstellt werden.

- Instanzattribute — Jedes Instanzattribut enthält eine Reihe von Attributen, die die Instanz charakterisieren, z. B. die private IP-Adresse und den privaten DNS-Namen der Instanz. Der Kürze halber zeigt das Beispiel nur das `php-app2`-Attribut im Detail. Die anderen enthalten ähnliche Informationen.
- `applications`— Eine Liste der bereitgestellten Apps, die in diesem Beispiel nicht verwendet wurden.
- `stack`— Der Stack-Name; `MyStack` in diesem Beispiel.
- `instance`— Die Instanz, auf der diese Attribute installiert sind; `php-app1` in diesem Beispiel. Rezepte können diese Attribute zum Abrufen von Informationen über die Instance nutzen, auf der sie ausgeführt werden, beispielsweise die öffentliche IP-Adresse der Instance.
- `activity`— Die Aktivität, die die Attribute erzeugt hat; in diesem Beispiel ein Configure-Ereignis.
- `rails_stack`— Der Rails-Stack für Stacks, die eine Rails App Server-Ebene enthalten.
- `deployment`— Ob diese Attribute mit einer Bereitstellung verknüpft sind. In diesem Beispiel auf `null` gesetzt, da die Attribute einem Konfigurationsereignis zugeordnet sind.
- `valid_client_activities`— Eine Liste gültiger Kundenaktivitäten.

Das `opsworks`-Attribut wird gefolgt von mehreren Attributen der oberen Ebene, einschließlich:

- `opsworks_custom_cookbooks`— Ob benutzerdefinierte Kochbücher aktiviert sind. Wenn dies der Fall ist, enthält das Attribut eine Liste benutzerdefinierter Rezepte.
- `recipes`— Die Rezepte, die im Rahmen dieser Aktivität ausgeführt wurden.
- `opsworks_rubygems`— Die RubyGems Version der Instanz.
- `ssh_users`— Eine Liste von SSH-Benutzern; in diesem Beispiel keine.
- `opsworks_bundler`— Die Bundler-Version und ob sie aktiviert ist.

- `deploy`— Informationen über Bereitstellungsaktivitäten; in diesem Beispiel keine.

Bereitstellungsattribute

Die Attribute für ein Bereitstellungsereignis oder den Stack-Befehl zum Ausführen von Rezepten bestehen aus den integrierten Attributen für die Stack-Konfiguration und -Bereitstellung sowie allen benutzerdefinierten Stack- oder Bereitstellungsattributen (hier keine). Das folgende JSON-Objekt zeigt die Attribute aus php-app1, die einem Bereitstellungsereignis zugeordnet sind, das die SimplePHP-App auf den PHP-Instances des Stacks bereitgestellt hat. Das Objekt besteht zum Großteil aus Stack-Konfigurationsattributen, die denen für das Konfigurationsereignis ähneln, das im vorherigen Abschnitt beschrieben wurde. Deshalb konzentriert sich dieses Beispiel primär auf bereitstellungsspezifische Attribute. Eine detaillierte Beschreibung verschiedener Attribute finden Sie unter Stack-Konfigurations- und Bereitstellungsattribute: Linux und Integrierte Rezeptbuchattribute.

```
{
   ...
  "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
      }
    }
  },
}
```

Das `opsworks`-Attribut ist nahezu identisch mit dem Beispiel aus dem vorherigen Abschnitt. Die folgenden Abschnitte sind primär für die Bereitstellung relevant:

- `activity`— Das Ereignis, das diesen Attributen zugeordnet ist; in diesem Beispiel ein Deploy-Ereignis.
- `applications`— Enthält eine Reihe von Attributen für jede App, die die Namen, Slug-Namen und Typen der Apps bereitstellen.

  Der Slug-Name ist der Kurzname der App, den AWS OpsWorks Stacks aus dem App-Namen generiert. Der Slug-Name für SimplePHP ist simplephp.
- `deployment`— Die Bereitstellungs-ID, die eine Bereitstellung eindeutig identifiziert.

Das `deploy`-Attribut enthält Informationen über die Apps, die bereitgestellt werden. So verwenden beispielsweise die integrierten Bereitstellungsrezepte die Daten des `deploy`-Attributs, um Dateien in den entsprechenden Verzeichnisses zu installieren und Datenbankverbindungsdateien zu erstellen. Das `deploy`-Attribut enthält ein Attribut für jede bereitgestellte App, wobei die Kurzbezeichnung der App verwendet wird. Jedes App-Attribut enthält die folgenden Attribute:

- `environment_variables`— Enthält alle Umgebungsvariablen, die Sie für die App definiert haben. Weitere Informationen finden Sie unter Umgebungsvariablen.
- `domains`— Standardmäßig ist die Domain der Kurzname der App, der in diesem Beispiel simplephpapp lautet. Wenn Sie benutzerdefinierte Domänen zugewiesen haben, werden diese hier ebenfalls angezeigt. Weitere Informationen finden Sie unter Verwenden von benutzerdefinierten Domänen.
- `application`— Der Kurzname der App.
- `scm`— Dieses Element enthält die Informationen, die zum Herunterladen der Dateien der App aus ihrem Repository erforderlich sind. In diesem Beispiel handelt es sich um ein Git-Repository.
- `database`— Datenbankinformationen, wenn der Stapel eine Datenbankschicht enthält.
- `document_root`— Das Dokumentenstammverzeichnis, das `null` in diesem Beispiel auf eingestellt ist, was darauf hinweist, dass das Stammverzeichnis öffentlich ist.
- `ssl_certificate_ca`,`ssl_support`, `ssl_certificate_key` — Gibt an, ob die App SSL-Unterstützung bietet. Wenn dies der Fall ist, werden die Attribute `ssl_certificate_key` und `ssl_certificate_ca` auf die entsprechenden Zertifikate gesetzt.

- `deploy_to`— Das Stammverzeichnis der App.

Rezeptbücher 101

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Ein AWS OpsWorks Stacks-Stack auf Produktionsebene erfordert in der Regel einige Anpassungen, was häufig die Implementierung eines benutzerdefinierten Chef-Kochbuchs mit einem oder mehreren Rezepten, Attributdateien oder Vorlagendateien bedeutet. Dieses Thema ist ein Tutorial zur Einführung in die Implementierung von Kochbüchern für Stacks. AWS OpsWorks

Weitere Informationen darüber, wie AWS OpsWorks Stacks Kochbücher verwendet, einschließlich einer kurzen allgemeinen Einführung in Kochbücher, finden Sie unter. Cookbooks und Rezepte Weitere Informationen zum Implementieren und Testen von Chef-Rezepten finden Sie in dem Buch Test-Driven Infrastructure with Chef, 2nd Edition.

Die Tutorial-Beispiele sind in zwei Abschnitte unterteilt:

- Rezeptbücher – Grundlagen ist eine Gruppe von Anleitungen für Benutzer, die keine Erfahrung im Umgang mit Chef haben. Erfahrene Chef-Benutzer können diesen Abschnitt überspringen.

  Die Beispiele erläutern Ihnen die Grundlagen zur Implementierung von Rezeptbüchern, um allgemeine Aufgaben wie z. B. das Installieren von Paketen oder Erstellen von Verzeichnissen durchzuführen. Zur Vereinfachung des Prozesses verwenden Sie zwei nützliche Tools, um die meisten Beispiele lokal auf einer virtuellen Maschine auszuführen: Vagrant und Test Kitchen. Bevor Sie beginnen, Rezeptbücher – Grundlagen, lesen Sie zuerst Vagrant und Test Kitchen, um zu erfahren, wie Sie diese Tools installieren und verwenden. Da Windows von Test Kitchen noch nicht unterstützt wird, gelten alle Beispiele für Linux (die Notizen geben an, wie dies für Windows angepasst werden kann).
- Implementierung von Kochbüchern für Stacks AWS OpsWorksbeschreibt, wie Rezepte für AWS OpsWorks Stacks implementiert werden, auch für Windows-Stacks.

Es enthält auch einige fortgeschrittenere Informationen, z. B. die Verwendung von Berkshelf zur Verwaltung externer Kochbücher. Die Beispiele richten sich an neue Chef-Benutzer, wie die Beispiele in Rezeptbücher – Grundlagen. AWS OpsWorks Stacks funktioniert jedoch etwas anders als der Chef-Server, daher empfehlen wir erfahrenen Chef-Benutzern, zumindest diesen Abschnitt durchzulesen.

Vagrant und Test Kitchen

Wenn Sie Rezepte für Linux-Instances anwenden, sind Vagrant und Test Kitchen sehr hilfreiche Tools zum Erlernen und für die erste Entwicklungs- und Testphase. Dies enthält kurze Beschreibungen von Vagrant und Test Kitchen und weist Sie auf Installationsanweisungen und Komplettlösungen hin, mit denen Sie die Tools einrichten und mit den Grundlagen der Verwendung der Tools vertraut machen können. Obwohl Windows von Vagrant unterstützt wird, ist dies bei Test Kitchen nicht der Fall, daher werden nur Linux-Beispiele für diese Tools erläutert.

Vagrant

Vagrant stellt eine konsistente Umgebung zur Ausführung und zum Testen von Code auf einer virtuellen Maschine zur Verfügung. Es unterstützt eine Vielzahl von Umgebungen — sogenannte Vagrant-Boxen —, von denen jede ein konfiguriertes Betriebssystem darstellt. Für AWS OpsWorks Stacks basieren die interessierenden Umgebungen auf Ubuntu-, Amazon- oder Red Hat Enterprise Linux (RHEL) -Distributionen, sodass in den Beispielen hauptsächlich eine Vagrant-Box mit dem Namen verwendet wird. `opscode-ubuntu-12.04`

Vagrant ist für Linux, Windows und Macintosh-Systeme verfügbar, sodass Sie Ihre bevorzugte Workstation verwenden können, um Rezepte auf allen unterstützten Betriebssystemen zu implementieren und zu testen. Die Beispiele für dieses Kapitel wurden auf einem Ubuntu-Linux-System erstellt, aber die Übersetzung der Verfahren auf Windows- oder Macintosh-Systeme ist einfach.

Vagrant ist im Wesentlichen ein Wrapper für einen Anbieter von Virtualisierungsdiensten. Die meisten Beispiele verwenden den Anbieter. VirtualBox VirtualBox ist kostenlos und für Linux-, Windows- und Macintosh-Systeme verfügbar. Die Vagrant-Komplettlösung enthält Installationsanweisungen, falls Sie diese noch nicht auf Ihrem System installiert haben VirtualBox . Beachten Sie, dass Sie auf Ubuntu basierende Umgebungen ausführen können VirtualBox, Amazon Linux jedoch nur für Amazon EC2 EC2-Instances verfügbar ist. Sie können jedoch ein ähnliches Betriebssystem wie CentOS ausführen VirtualBox, was für die anfängliche Entwicklung und das Testen nützlich ist.

Weitere Informationen zu anderen Anbietern finden Sie in der Vagrant-Dokumentation. Insbesondere ermöglicht Ihnen der `vagrant-aws` Plug-in-Anbieter die Verwendung von Vagrant mit Amazon EC2 EC2-Instances. Dieser Anbieter ist besonders nützlich, um Rezepte auf Amazon Linux zu testen, das nur auf Amazon EC2 EC2-Instances verfügbar ist. Der `vagrant-aws`-Anbieter ist kostenlos. Sie benötigen jedoch ein AWS-Konto und es werden die von Ihnen verwendeten AWS-Ressourcen berechnet.

An dieser Stelle empfehlen wir Ihnen die Anleitung Getting Started von Vagrant, die Ihnen erläutert, wie Sie Vagrant auf Ihrer Workstation installieren, und die Ihnen die Grundlagen zur Verwendung von Vagrant vermittelt. Beachten Sie, dass die Beispiele in diesem Kapitel kein Git-Repository verwenden, sodass Sie diesen Teil der Anleitung überspringen können.

Test Kitchen

Test Kitchen vereinfacht die Ausführung und das Testen Ihrer Rezeptbücher auf Vagrant. In der Praxis werden Sie Vagrant nur in seltenen Fällen direkt verwenden müssen. Test Kitchen führt die gängigsten Aufgaben aus, darunter:

- Starten einer Instance in Vagrant.
- Übertragen von Rezeptbüchern auf die Instance.
- Ausführen der Rezepte des Rezeptbuchs in der Instance.
- Testen eines Rezepts des Rezeptbuchs in der Instance.
- Verwenden von SSH für die Anmeldung bei der Instance.

Anstelle der direkten Installation des Test Kitchen-Gems empfehlen wir, Chef DK zu installieren. Neben Chef selbst enthält dieses Paket Test Kitchen, Berkshelf und mehrere andere nützliche Tools. ChefSpec

An dieser Stelle sollten Sie die Anleitung Getting Started von Test Kitchen durcharbeiten. Hier werden Ihnen die Grundlagen vermittelt, wie Sie Test Kitchen zum Ausführen und Testen von Rezepten verwenden.

> **Note**
>
> In den in diesem Kapitel aufgeführten Beispielen wird Test Kitchen als eine praktische Methode für die Ausführung von Rezepten verwendet. Wenn Sie möchten, können Sie die Anleitung "Erste Schritte" unterbrechen, nachdem Sie den Abschnitt "Manuelles Überprüfen"

abgeschlossen haben, in dem alle wesentlichen Informationen für die Beispiele enthalten sind. Test Kitchen ist jedoch in erster Linie eine Test-Plattform, die Test-Frameworks wie das Bash-automatisierte Testsystem (BATS) unterstützt. Gehen Sie den Rest der Anleitung zu einem späteren Zeitpunkt durch, um zu erfahren, wie Sie Test Kitchen zum Testen Ihrer Rezepte verwenden können.

Rezeptbücher – Grundlagen

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können Rezeptbücher verwenden, um zahlreiche Aufgaben auszuführen. In den folgenden Themen wird davon ausgegangen, dass Sie mit Chef nicht vertraut sind. Daher wird beschrieben, wie Sie Rezeptbücher zur Ausführung zahlreicher gängiger Aufgaben verwenden können. Da Windows von Test Kitchen noch nicht unterstützt wird, gelten alle Beispiele für Linux (die Notizen geben an, wie dies für Windows angepasst werden kann). Sofern Sie mit Chef noch nicht vertraut sind, wird empfohlen, dass Sie diese Beispiele durcharbeiten (auch, wenn Sie Windows nutzen). Die meisten Beispiele in diesem Thema lassen sich mit geringfügigen Änderungen (die in den Beispielen angegeben werden) auch für Windows-Instances nutzen. Alle Beispiele werden auf einer virtuellen Maschine ausgeführt, daher benötigen Sie keinen Linux-Computer. Installieren Sie einfach Vagrant und Test Kitchen auf Ihrer regulären Workstation.

Note

Falls Sie diese Rezepte auf einer Windows-Instance ausführen möchten, ist es am einfachsten, einen Windows-Stack zu erstellen und die Rezepte auf einer der Stack-Instances auszuführen. Weitere Informationen zum Ausführen von Rezepten auf einer AWS OpsWorks Stacks-Windows-Instanz finden Sie unter. Ausführen eines Rezepts auf einer Windows-Instance

Bevor Sie fortfahren, stellen Sie sicher, dass Sie Vagrant und Test Kitchen installiert und die jeweiligen Anleitungen für die ersten Schritte durchgearbeitet haben. Weitere Informationen finden Sie unter Vagrant und Test Kitchen.

Themen

Rezeptstruktur

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Ein Rezeptbuch ist in erster Linie eine Sammlung von Rezepten, mit denen zahlreiche Aufgaben auf einer Instance ausgeführt werden können. Um die Implementierung von Rezepten zu verdeutlichen, ist ein einfaches Beispiel hilfreich. Im Folgenden finden Sie das Einrichtungsrezept für die integrierte HAProxy-Ebene. Konzentrieren Sie sich zum jetzigen Zeitpunkt nur auf die allgemeine Struktur. Machen Sie sich keine Gedanken über die Details, sie werden in den nachfolgenden Beispielen veranschaulicht.

```
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**
>
> Dieses und weitere Beispiele für funktionierende Rezepte und die zugehöriger Dateien finden Sie unter AWS OpsWorks Stacks built-in recipes.

In diesem Beispiel werden die wichtigsten Rezeptelemente vorgestellt, die in den folgenden Abschnitten beschrieben werden.

Themen

- Ressourcen
- Flusssteuerung

- Eingebundene Rezepte

Ressourcen

Rezepte bestehen zum Großteil aus Chef-Ressourcen. Jede gibt einen bestimmten Aspekt des letzten Instance-Status an, z. B. ein zu installierendes Paket oder ein zu startender Service. Im Beispiel werden vier Ressourcen verwendet:

- Eine `package`-Ressource, die für ein installiertes Paket steht – in diesem Beispiel ein HAProxy-Server.
- Eine `service`-Ressource, die für einen Service steht – in diesem Beispiel der HAProxy-Service.
- Zwei `template`-Ressourcen, die für aus einer bestimmten Vorlage zu erstellende Dateien stehen – in diesem Beispiel zwei HAProxy-Konfigurationsdateien.

Ressourcen sind eine deklarative Möglichkeit, um den Instance-Status anzugeben. Im Hintergrund ist jeder Ressource ein Anbieter zugeordnet, von dem die erforderlichen Aufgaben ausgeführt werden, z. B. Pakete installieren, Verzeichnisse erstellen und konfigurieren und Services starten. Falls die Details der Aufgabe vom jeweiligen Betriebssystem abhängen, verfügt die Ressource über mehrere Anbieter, von denen jeweils der geeignete für das System ausgewählt wird. Bei einem Red Hat Linux-System wird vom `package`-Anbieter `yum` zum Installieren der Pakete verwendet. Auf einem Ubuntu Linux-System verwendet der `package`-Anbieter hingegen `apt-get`.

Eine Ressource wird als Ruby-Codeblock im folgenden allgemeinen Format implementiert.

```
resource_type "resource_name" do
  attribute1 'value1'
  attribute2 'value2'
  ...
  action :action_name
  notifies : action 'resource'
end
```

Die Elemente lauten folgendermaßen:

Ressourcentyp

(Erforderlich) Das Beispiel enthält drei Ressourcentypen, nämlich `package`, `service` und `template`.

Ressourcenname

(Erforderlich) Der Name identifiziert eine bestimmte Ressource und wird gelegentlich als Standardwert für eines der Attribute verwendet. In diesem Beispiel steht `package` für eine "package"-Ressource mit dem Namen `haproxy` und die erste `template`-Ressource steht für eine Konfigurationsdatei mit dem Namen `/etc/default/haproxy`.

Attribute

(Optional) Attribute geben die Ressourcenkonfiguration an. Sie hängen vom Ressourcentyp und davon, wie Sie die Ressource konfigurieren möchten, ab.

- Im Beispiel definieren die `template`-Ressourcen explizit mehrere Attribute, die jeweils die Quelle, den Besitzer, die Gruppe und den Modus der erstellten Datei spezifizieren.
- Von den im Beispiel verwendeten Ressourcen `package` und `service` werden keine Attribute explizit definiert.

  Der Ressourcenname ist in der Regel der Standardwert für ein erforderliches Attribut – und häufig ist auch nicht mehr nötig. Beispielsweise ist der Ressourcenname der Standardwert für das `package`-Attribut der `package_name`-Ressource und gleichzeitig auch das einzig erforderliche Attribut.

Die so genannten "Wächterattribute" sind besondere Attribute und geben an, wann eine Aktion seitens des Ressourcenanbieters erforderlich ist. Beispielsweise fordert das `only_if`-Attribut den Ressourcenanbieter nur zu einer Aktion auf, sofern eine festgelegte Bedingung erfüllt wird. Im HAProxy-Rezept werden keine Wächterattribute genutzt, aber sie werden in einigen der folgenden Beispiele verwendet.

Aktionen und Benachrichtigungen

(Optional) Aktionen und Benachrichtigungen geben an, welche Aufgaben vom Anbieter ausgeführt werden sollen.

- Mit `action` wird der Anbieter zu einer bestimmten Aktion aufgefordert, z. B. etwas zu installieren oder zu erstellen.

  Für jede Ressource sind mehrere ressourcenabhängige Aktionen möglich, eine davon ist immer die Standardaktion. In diesem Beispiel lautet die Aktion für die `package`-Ressource `install` und weist den Anbieter an, das Paket zu installieren. Die erste `template`-Ressource hat kein `action`-Element, daher führt der Anbieter die `create`-Standardaktion aus.
- Mit `notifies` wird der Anbieter einer anderen Ressource zur Ausführung einer Aktion aufgefordert. Dies gilt nur, wenn sich der Ressourcenstatus geändert hat.

`notifies` wird in der Regel mit Ressourcen wie `template` und `file` für die Aufgabenausführung verwendet, z. B. um den Service nach einer Änderung der Konfigurationsdatei neu zu starten. Ressourcen verfügen nicht über Standardbenachrichtigungen. Wenn eine Benachrichtigung gesendet werden soll, muss für die Ressource explizit ein `notifies`-Element deklariert werden. Im HAProxy-Rezept benachrichtigt die zweite `template`-Ressource die haproxy `service`-Ressource über den Neustart des HAProxy-Service nach einer Änderung der zugehörigen Konfigurationsdatei.

Manchmal hängen Ressourcen vom Betriebssystem ab.

- Einige Ressourcen können nur auf Linux- oder Windows-Systemen verwendet werden.

  Beispielsweise werden mit package Pakete auf Linux-Systemen und mit windows_package Pakete auf Windows-Systemen installiert.
- Einige Ressourcen können mit einem beliebigen Betriebssystem genutzt werden, haben aber Attribute für ein bestimmtes System.

  Beispielsweise kann die file-Ressource sowohl auf Linux- als auch auf Windows-Systemen eingesetzt werden, verfügt aber über unterschiedliche Attributsätze für die Berechtigungskonfiguration.

Beschreibungen der Standardressourcen einschließlich der verfügbaren Attribute, Aktionen und Benachrichtigungen für die einzelnen Ressourcen finden Sie unter About Resources and Providers.

Flusssteuerung

Da es sich bei Rezepten um Ruby-Anwendungen handelt, können Sie Ruby-Steuerungsstrukturen für die Einbindung der Flusssteuerung in ein Rezept verwenden. Beispielsweise können Sie mit der Ruby-Bedingungslogik unterschiedliches Rezeptverhalten auf verschiedenen Systemen erzeugen. Das HAProxy-Rezept enthält einen `if`-Block, der mithilfe einer `template`-Ressource eine Konfigurationsdatei erstellt, vorausgesetzt, das Rezept wird auf einem Debian- oder Ubuntu-System ausgeführt.

Ein anderes gängiges Szenario besteht darin, in einer Schleife eine Ressource mehrere Male mit unterschiedlichen Attributeinstellungen auszuführen. Beispielsweise können Sie Verzeichnisse anlegen, indem Sie eine `directory`-Ressource mehrfach mit unterschiedlichen Verzeichnisnamen in einer Schleife ausführen.

> **Note**
>
> Falls Sie mit Ruby nicht vertraut sind, finden Sie die für die meisten Rezepte erforderlichen Informationen unter Just Enough Ruby for Chef.

Eingebundene Rezepte

Mit `include_recipe` können Sie weitere Rezepte in den Code einbinden, sodass Sie die Rezepte "modularisieren" und denselben Code in mehreren Rezepten verwenden können. Vor der Ausführung des Host-Rezepts ersetzt Chef jedes `include_recipe`-Element durch den angegebenen Rezeptcode. Sie erkennen ein eingebundenes Rezept an der Standardsyntax von Chef, `cookbook_name::recipe_name`, wobei für `recipe_name` die Erweiterung `.rb` fehlt. Im Beispiel ist das Rezept `haproxy::service` für den HAProxy-Service enthalten.

> **Note**
>
> Falls Sie Rezepte aus einem anderen Rezeptbuch mit `include_recipe` in Rezepte einbinden, die mit Chef 11.10 und neuer ausgeführt werden, müssen Sie in einer `depends`-Anweisung die Abhängigkeit in der Datei `metadata.rb` des Rezeptbuchs deklarieren. Weitere Informationen finden Sie unter Implementieren von Rezepten: Chef 11.10.

Beispiel 1: Installieren von Paketen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Die Paketinstallation zählt zu den gängigeren Einsatzzwecken von Rezepten und kann, je nach Paket, recht einfach sein. Beispielsweise wird mit folgendem Paket Git auf einem Linux-System installiert.

```
package 'git' do
  action :install
end
```

Die Paketinstallation wird von der package-Ressource ausgeführt. In diesem Beispiel müssen keine Attribute angegeben werden. Der Ressourcenname ist der Standardwert für das `package_name`-Attribut, mit dem das Paket identifiziert wird. Mit der `install`-Aktion wird der Anbieter aufgefordert, das Paket zu installieren. Sie könnten den Code noch weiter vereinfachen, indem Sie `install` weglassen, denn das ist die Standardaktion der `package`-Ressource. Wenn Sie das Rezept ausführen, wird von Chef der entsprechende Anbieter für die Paketinstallation verwendet. In diesem Beispiel wird ein Ubuntu-System verwendet, auf dem der Anbieter Git durch den Aufruf von `apt-get` installiert.

> **Note**
>
> Bei der Softwareinstallation auf einem Windows-System ist ein anderes Vorgehen erforderlich. Weitere Informationen finden Sie unter Installieren von Windows-Software.

Wenn Sie dieses Rezept in Vagrant mit Test Kitchen ausführen möchten, müssen Sie zunächst ein Rezeptbuch einrichten und Test Kitchen initialisieren und konfigurieren. Die nachfolgenden Schritte gelten für ein Linux-System, aber das Verfahren ist für Windows- und Macintosh-Systeme im Wesentlichen gleich. Öffnen Sie zunächst ein Terminalfenster. In allen Beispielen dieses Kapitels werden Befehlszeilen-Tools genutzt.

So bereiten Sie ein Rezeptbuch vor

1. Erstellen Sie in Ihrem Stammverzeichnis das Unterverzeichnis `opsworks_cookbooks`, das alle Rezeptbücher für dieses Kapitel enthalten wird. Erstellen Sie anschließend ein Unterverzeichnis mit dem Namen `installpkg` für dieses Rezeptbuch und öffnen Sie es.
2. Erstellen Sie in `installpkg` die Datei `metadata.rb`, die folgenden Code enthält.

   ```
   name "installpkg"
   version "0.1.0"
   ```

Aus Gründen der Übersichtlichkeit werden in den Beispielen dieses Kapitels nur der Name und die Version des Rezeptbuchs angegeben, aber `metadata.rb` kann eine Vielzahl von Metadaten für ein Rezeptbuch enthalten. Weitere Informationen finden Sie unter About Cookbook Metadata.

> **Note**
>
> Achten Sie darauf, `metadata.rb` vor der Test Kitchen-Initialisierung zu erstellen, denn die Daten werden für die Standardkonfigurationsdatei benötigt.

3. Führen Sie in `installpkg` den Befehl `kitchen init` zur Test Kitchen-Initialisierung und zur Installation des Vagrant-Standardtreibers aus.
4. Mit dem Befehl `kitchen init` wird in `installpkg` eine YAML-Konfigurationsdatei mit dem Namen `.kitchen.yml` generiert. Öffnen Sie die Datei in Ihrem bevorzugten Texteditor. Die Datei `.kitchen.yml` enthält einen `platforms`-Bereich mit den Systemen, auf denen die Rezepte ausgeführt werden sollen. Test Kitchen generiert eine Instance und führt die spezifizierten Rezepte auf den einzelnen Plattformen aus.

> **Note**
>
> Standardmäßig führt Test Kitchen die Rezepte nur auf jeweils einer Plattform aus. Wenn Sie ein `-p`-Argument zu den Befehlen hinzufügen, mit denen die Instance erstellt wird, führt Test Kitchen die Rezepte auf allen Plattformen gleichzeitig aus.

Eine einzelne Plattform ist für dieses Beispiel ausreichend. Bearbeiten Sie daher `.kitchen.yml` und entfernen Sie die `centos-6.4`-Plattform. Ihre Datei `.kitchen.yml` sollte nun wie folgt aussehen:

```
---
driver:
  name: vagrant

provisioner:
  name: chef_solo

platforms:
  - name: ubuntu-12.04
```

```
suites:
  - name: default
    run_list:
      - recipe[installpkg::default]
    attributes:
```

Test Kitchen führt nur die Rezepte aus, die in der `.kitchen.yml`-Ausführungsliste genannt werden. Sie erkennen die Rezepte am Format `[`*`cookbook_name`*`::`*`recipe_name`*`]`, wobei für *`recipe_name`* die Erweiterung `.rb` fehlt. Anfänglich enthält die `.kitchen.yml`-Ausführungsliste das Standardrezept des Rezeptbuchs, `installpkg::default`. Dieses Rezept werden Sie implementieren, daher muss die Ausführungsliste nicht geändert werden.

5. Erstellen Sie ein Unterverzeichnis von `installpkg` namens `recipes`.

   Wenn ein Kochbuch Rezepte enthält — was bei den meisten der Fall ist —, müssen sie sich im Unterverzeichnis befinden. `recipes`

Nun können Sie das Rezept zum Rezeptbuch hinzufügen und mithilfe von Test Kitchen auf einer Instance ausführen.

So führen Sie das Rezept aus

1. Erstellen Sie eine Datei mit dem Namen `default.rb`, fügen Sie den Beispiel-Code für die Git-Installation vom Abschnittsanfang ein und speichern Sie die Datei im Unterverzeichnis `recipes`.
2. Führen Sie im Verzeichnis `installpkg` den Befehl `kitchen converge` aus. Dieser Befehl startet eine neue Ubuntu-Instanz in Vagrant, kopiert Ihre Kochbücher auf die Instanz und initiiert einen Chef-Lauf, um die Rezepte in der Ausführungsliste auszuführen. `.kitchen.yml`
3. Um zu prüfen, ob das Rezept erfolgreich ausgeführt wurde, führen Sie `kitchen login` aus und öffnen damit eine SSH-Verbindung zur Instance. Führen Sie dann `git --version` aus, um zu überprüfen, ob Git erfolgreich installiert wurde. Um zur Workstation zurückzukehren, führen Sie `exit` aus.
4. Wenn Sie fertig sind, führen Sie `kitchen destroy` aus und fahren damit die Instance herunter. Im nächsten Beispiel wird ein anderes Rezeptbuch verwendet.

Dieses Beispiel ist gut für die ersten Schritte geeignet, es ist jedoch besonders einfach. Die Installation anderer Pakete kann komplizierter sein und Sie müssen möglicherweise einen oder alle der folgenden Schritte ausführen:

- Erstellen und konfigurieren Sie einen Benutzer.
- Erstellen Sie ein oder mehrere Verzeichnisse für Daten, Protokolle usw.
- Installieren Sie eine oder mehrere Konfigurationsdateien.
- Geben Sie einen anderen Paketnamen oder verschiedene Attributwerte für unterschiedliche Betriebssysteme an.
- Starten Sie einen Service und starten Sie diesen bei Bedarf neu.

In den folgenden Beispielen wird erklärt, wie Sie mit diesen Aspekten umgehen. Zudem werden einige weitere hilfreiche Vorgehensweisen beschrieben.

Beispiel 2: Verwalten von Benutzern

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Eine andere einfache Aufgabe ist das Verwalten von Benutzern auf einer Instance. Mit dem folgenden Rezept wird ein neuer Benutzer zu einer Linux-Instance hinzugefügt.

```
user "myuser" do
  home "/home/newuser"
  shell "/bin/bash"
end
```

Verwenden Sie eine user-Ressource, um die Benutzer sowohl auf Linux- als auch auf Windows-Systemen zu verwalten. Einige Attribute gelten jedoch nur für ein System. Im Beispiel wird der Benutzer `myuser` erstellt, für den Stammverzeichnis und Shell angegeben werden. Es ist keine Aktion vorgegeben, daher wird von der Ressource die `create`-Standardaktion verwendet. Sie können Attribute zu `user` hinzufügen und so weitere Einstellungen (z. B. Passwort oder Gruppen-ID) spezifizieren. Zudem können Sie `user` für entsprechende Benutzerverwaltungsaufgaben einsetzen,

z. B. Benutzereinstellungen ändern oder Benutzer löschen. Weitere Informationen finden Sie unter user.

So führen Sie das Rezept aus

1. Erstellen Sie ein Verzeichnis in `opsworks_cookbooks` namens `newuser` und öffnen Sie es.
2. Erstellen Sie die Datei `metadata.rb`, die folgenden Code enthält, und speichern Sie diese unter `newuser`.

   ```
   name "newuser"
   version "0.1.0"
   ```

3. Initialisieren und konfigurieren Sie Test Kitchen wie unter Beispiel 1: Installieren von Paketen beschrieben und fügen Sie das Verzeichnis `recipes` zum Verzeichnis `newuser` hinzu.
4. Fügen Sie die Datei `default.rb` mit dem Beispielrezept zum Rezeptbuch-Verzeichnis `recipes` hinzu.
5. Führen Sie `kitchen converge` aus, um das Rezept auszuführen.
6. Melden Sie sich über `kitchen login` an der Instance an. Prüfen Sie, ob der neue Benutzer vorhanden ist, indem Sie `cat /etc/passwd` ausführen. Der Benutzer `myuser` sollte unten in der Datei angezeigt werden.

Beispiel 3: Erstellen von Verzeichnissen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn Sie ein Paket auf einer Instance installieren, müssen Sie häufig einige Konfigurationsdateien erstellen und sie in den entsprechenden Verzeichnissen platzieren. Doch diese Verzeichnisse sind möglicherweise noch nicht vorhanden. Zudem müssen ggf. auch Verzeichnisse für Daten, Protokolldateien usw. erstellt werden. Beispielsweise booten Sie zuerst das Ubuntu-System, das Sie für die meisten Beispiele verwenden. Das `/srv` Verzeichnis hat keine Unterverzeichnisse. Wenn Sie

einen Anwendungsserver installieren, benötigen Sie das Verzeichnis `/srv/www/` und vermutlich auch einige Unterverzeichnisse für Datendateien, Protokolle und so weiter. Mit dem folgenden Rezept wird `/srv/www/` auf einer Instance erstellt.

```
directory "/srv/www/" do
  mode 0755
  owner 'root'
  group 'root'
  action :create
end
```

Mithilfe einer `directory`-Ressource erstellen und konfigurieren Sie Verzeichnisse auf Linux- und Windows-Systemen, wobei einige Attribute unterschiedlich verwendet werden. Der Ressourcenname ist der Standardwert für das `path`-Attribut der Ressource, daher wird im Beispiel das Verzeichnis `/srv/www/` mit den Eigenschaften `mode`, `owner` und `group` erstellt.

So führen Sie das Rezept aus

1. Erstellen Sie ein Verzeichnis in `opsworks_cookbooks` namens `createdir` und öffnen Sie es.
2. Initialisieren und konfigurieren Sie Test Kitchen wie unter Beispiel 1: Installieren von Paketen beschrieben und fügen Sie das Verzeichnis `recipes` zu `createdir` hinzu.
3. Fügen Sie die Datei `default.rb` mit dem Rezeptcode zum Rezeptbuch-Unterverzeichnis `recipes` hinzu.
4. Führen Sie `kitchen converge` aus, um das Rezept auszuführen.
5. Führen Sie `kitchen login` aus und öffnen Sie `/srv`, um zu prüfen, ob das Unterverzeichnis `www` vorhanden ist.
6. Führen Sie `exit` aus, um zur Workstation zurückzukehren, und lassen Sie die Instance aktiv.

> **Note**
>
> Um auf der Instance ein Verzeichnis ähnlich dem Stammverzeichnis zu erstellen, bilden Sie das Stammverzeichnis mit `#{ENV['HOME']}` ab. Beispielsweise wird wie folgt das Verzeichnis `~/shared` erstellt.
>
> ```
> directory "#{ENV['HOME']}/shared" do
> ```

```
  ...
end
```

Angenommen, Sie möchten ein tiefer geschachteltes Verzeichnis wie `/srv/www/shared` erstellen. Dann modifizieren Sie das vorherige Rezept wie folgt.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  action :create
end
```

So führen Sie das Rezept aus

1. Ersetzen Sie den Code in `default.rb` durch das vorherige Rezept.
2. Führen Sie `kitchen converge` im Verzeichnis `createdir` aus.
3. Überprüfen Sie, ob das Verzeichnis erstellt wurde. Führen Sie dazu `kitchen login` aus und öffnen Sie `/srv/www`, um zu prüfen, ob das Unterverzeichnis `shared` vorhanden ist.
4. Führen Sie `kitchen destroy` aus, um die Instance herunterzufahren.

Wie Sie sehen können, wurde der Befehl `kitchen converge` viel schneller ausgeführt. Das liegt daran, dass die Instance bereits ausgeführt wird, daher ist es nicht nötig, die Instance zu starten, Chef zu installieren usw. Test Kitchen kopiert einfach das aktualisierte Rezeptbuch auf die Instance und startet Chef.

Führen Sie nun `kitchen converge` noch einmal aus, damit das Rezept auf einer neuen Instance ausgeführt wird. Das Ergebnis sieht folgendermaßen aus.

```
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]
>>>>>> ----------------------
```

Was ist passiert? Das Problem ist, dass mit einer `directory`-Ressource standardmäßig nur ein Verzeichnis – und nicht mehrere – erstellt werden kann. Das Rezept konnte zuvor erfolgreich ausgeführt werden, weil das zuerst auf der Instance ausgeführte Rezept das Verzeichnis `/srv/www` bereits erstellt hatte, folglich wurde mit `/srv/www/shared` nur ein Unterverzeichnis erstellt.

> **Note**
>
> Achten Sie beim Ausführen von `kitchen converge` darauf, ob Sie die Rezepte auf einer neuen oder einer vorhandenen Instance ausführen. Die Ergebnisse könnten unterschiedlich ausfallen.

Um mehrere Unterverzeichnisse zu erstellen, fügen Sie zu `recursive` das `directory`-Attribut mit dem Wert `true` hinzu. Mit dem folgenden Rezept wird `/srv/www/shared` direkt auf einer neuen Instance erstellt.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end
```

Beispiel 4: Hinzufügen der Flusssteuerung

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Einige Rezepte sind nur eine Reihe von Chef-Ressourcen. In dem Fall werden bei der Rezeptausführung einfach die einzelnen Ressourcenanbieter nacheinander ausgeführt. Allerdings ist ein komplexerer Ausführungspfad meist sinnvoller. Nachfolgend finden Sie zwei gängige Szenarien:

- Ein Rezept soll die gleiche Ressource mehrfach und mit unterschiedlichen Attributeinstellungen ausführen.
- Für unterschiedliche Betriebssysteme sollen verschiedene Attributeinstellungen verwendet werden.

Sie können solche Szenarien durch die Einbindung von Ruby-Steuerungsstrukturen in das Rezept realisieren. In diesem Abschnitt wird erklärt, wie Sie das Rezept aus Beispiel 3: Erstellen von Verzeichnissen für beide Szenarien anpassen.

Themen

- Iteration
- Bedingungslogik

Iteration

In Beispiel 3: Erstellen von Verzeichnissen wurde veranschaulicht, wie Sie mit einer `directory`-Ressource ein oder mehrere Verzeichnisse erstellen. Aber was ist, wenn Sie zwei separate Verzeichnisse – `/srv/www/config` und `/srv/www/shared` – erstellen möchten? Sie können für jedes Verzeichnis eine separate "directory"-Ressource implementieren. Sollen viele Verzeichnisse erstellt werden, ist das jedoch sehr mühselig. Das folgende Rezept bietet dafür eine einfachere Methode.

```
[ "/srv/www/config", "/srv/www/shared" ].each do |path|
  directory path do
    mode 0755
    owner 'root'
    group 'root'
    recursive true
    action :create
  end
```

```
end
```

Anstatt für jedes Unterverzeichnis eine separate "directory"-Ressource zu verwenden, wird im Rezept eine Zeichenfolgensammlung mit enthaltenen Unterverzeichnispfaden genutzt. Bei der `each`-Methode von Ruby wird die Ressource einmal für jedes Sammlungselement (beginnend mit dem ersten) ausgeführt. Der Elementwert wird in der Ressource durch die `path`-Variable – in diesem Fall der Verzeichnispfad – dargestellt. Dieses Beispiel können Sie einfach anpassen, um eine beliebige Anzahl an Unterverzeichnissen zu erstellen.

So führen Sie das Rezept aus

1. Bleiben Sie im Verzeichnis `createdir`. Dieses Rezeptbuch wird auch in den nächsten Beispielen verwendet.
2. Führen Sie `kitchen destroy` aus, damit Sie mit einer neuen Instance beginnen können (sofern noch nicht geschehen).
3. Ersetzen Sie den Code in `default.rb` durch den Beispiel-Code und führen Sie `kitchen converge` aus.
4. Melden Sie sich an der Instance an. Die neu erstellten Verzeichnisse werden unter `/srv` angezeigt.

Sie können mithilfe einer Hash-Tabelle zwei Werte für jede Iteration angeben. Mit dem folgenden Rezept werden `/srv/www/config` und `/srv/www/shared` jeweils mit einem anderen Modus erstellt.

```
{ "/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
```

So führen Sie das Rezept aus

1. Führen Sie `kitchen destroy` aus, damit Sie mit einer neuen Instance beginnen können (sofern noch nicht geschehen).
2. Ersetzen Sie den Code in `default.rb` durch den Beispiel-Code und führen Sie `kitchen converge` aus.
3. Melden Sie sich an der Instance an. Die neu erstellten Verzeichnisse werden unter `/srv` mit den angegebenen Modi angezeigt.

> **Note**
>
> AWS OpsWorks In Stacks-Rezepten wird dieser Ansatz häufig verwendet, um Werte aus der JSON-Datei für die Stack-Konfiguration und Bereitstellung zu extrahieren — was im Grunde eine große Hashtabelle ist — und sie in eine Ressource einzufügen. Ein Beispiel finden Sie unter Bereitstellungsrezepte.

Bedingungslogik

Mithilfe der Bedingungslogik von Ruby können Sie auch mehrere Ausführungsvarianten erstellen. Im folgenden Rezept wird die Logik `if-elsif-else` als Erweiterung des vorherigen Beispiels eingesetzt, um das Unterverzeichnis `/srv/www/shared` zu erstellen, sofern es sich um Debian- und Ubuntu-Systeme handelt. Auf allen anderen Systemen wird in der Test Kitchen-Ausgabe eine Fehlermeldung protokolliert.

```
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
```

So führen Sie das Beispielrezept aus

1. Falls die Instance noch aktiv ist, fahren Sie sie mit `kitchen destroy` herunter.
2. Ersetzen Sie den Code in `default.rb` durch den Beispiel-Code.
3. Bearbeiten Sie `.kitchen.yml` und fügen Sie das CentOS 6.4-System zur Liste der Plattformen hinzu. Der `platforms`-Abschnitt der Datei sieht nun aus wie folgt.

```
...
platforms:
  - name: ubuntu-12.04
  - name: centos-6.4
...
```

4. Führen Sie `kitchen converge` aus, um eine Instance zu erstellen und die Rezepte für die einzelnen Plattformen in `.kitchen.yml` nacheinander auszuführen.

> **Note**
>
> Wenn nur eine Instance konvergiert werden soll, können Sie den Instance-Namen als Parameter hinzufügen. Um beispielsweise das Rezept nur auf der Ubuntu-Plattform zu konvergieren, führen Sie `kitchen converge default-ubuntu-1204` aus. Falls Sie die Namen der Plattformen vergessen haben, führen Sie einfach `kitchen list` aus.

Die Protokollmeldung im CentOS-Abschnitt der Test Kitchen-Ausgabe sieht in etwa folgendermaßen aus:

```
...
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
```

Nun können Sie sich an den Instances anmelden und prüfen, ob die Verzeichnisse erstellt wurden. Allerdings können Sie hier nicht einfach `kitchen login` ausführen. Sie müssen unter Angabe des Plattformnamens die Instance angeben, z. B. `kitchen login default-ubuntu-1204`.

> **Note**
>
> Sofern ein Test Kitchen-Befehl den Instance-Namen übernimmt, müssen Sie nicht den vollständigen Namen eingeben. Test Kitchen behandelt den Instance-Namen als regulären Ruby-Ausdruck, daher müssen nur genügend Zeichen eingegeben werden, um einen eindeutigen Treffer zu finden. Beispielsweise können Sie durch Ausführen von `kitchen converge ub` nur die Ubuntu-Instance konvergieren oder sich durch Ausführen von `kitchen login 64` an der CentOS-Instance anmelden.

Möglicherweise stellen Sie sich jetzt die Frage, woher das Rezept wissen kann, auf welcher Plattform es ausgeführt wird. Chef führt das Tool Ohai bei jeder Ausführung aus und erfasst so Systemdaten, darunter auch die Plattform. Diese Daten werden als Attribute in einer Struktur mit der Bezeichnung Knotenobjekt dargestellt. Mit der `platform?`-Methode von Chef werden die in Klammern gesetzten Systeme mit den Ohai-Plattformwerten verglichen. Bei einer Übereinstimmung wird der Wert "true" zurückgegeben.

Sie können den Wert eines Knotenattributs mit `node['attribute_name']` direkt im Code referenzieren. Der Plattformwert wird beispielsweise mit `node['platform']` angegeben. Sie könnten z. B. das vorherige Beispiel wie folgt schreiben.

```
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
```

Mit der Einbindung der Bedingungslogik in ein Rezept wird häufig die Tatsache berücksichtigt, dass verschiedene Linux-Familien gelegentlich unterschiedliche Namen für Pakete, Verzeichnisse etc.

verwenden. Beispielsweise lautet der Apache-Paketname auf CentOS-Systemen `httpd` und auf Ubuntu-Systemen `apache2`.

Falls Sie nur eine andere Zeichenfolge für verschiedene Systeme benötigen, ist die `value_for_platform`-Methode von Chef eine einfachere Lösung als `if-elsif-else`. Mit dem folgenden Rezept wird das Verzeichnis `/srv/www/shared` auf CentOS-Systemen, das Verzeichnis `/srv/www/data` auf Ubuntu-Systemen und das Verzeichnis `/srv/www/config` auf allen anderen Systemen erstellt.

```
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` weist den entsprechenden Pfad für `data_dir` zu und die `directory`-Ressource nutzt diesen Wert für die Verzeichniserstellung.

So führen Sie das Beispielrezept aus

1. Falls die Instance noch aktiv ist, fahren Sie sie mit `kitchen destroy` herunter.
2. Ersetzen Sie den Code in `default.rb` durch den Beispiel-Code.
3. Führen Sie `kitchen converge` aus und melden Sie sich anschließend an den einzelnen Instances an, um zu prüfen, ob die entsprechenden Verzeichnisse vorhanden sind.

Beispiel 5: Verwenden von Attributen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Für die Rezepte in den vorherigen Abschnitten wurden stets fest programmierte Werte verwendet, außer für die Plattform. Diese Methode kann ungünstig sein, z. B. wenn Sie denselben Wert in mehreren Rezepten verwenden möchten. Sie können Werte getrennt von Rezepten definieren, indem Sie eine Attributdatei in das Rezeptbuch einbinden.

Eine Attributdatei ist eine Ruby-Anwendung, mit der Werte für ein oder mehrere Attribute zugewiesen werden. Die Datei muss im Rezeptbuch-Ordner `attributes` sein. Chef bindet die Attribute in das Knotenobjekt ein, sodass alle Rezepte diese Attributwerte durch Referenzierung des Attributs verwenden können. In diesem Thema wird gezeigt, wie Sie das Rezept aus Iteration für die Nutzung von Attributen anpassen. Hier ist zu Referenzzwecken das ursprüngliche Rezept.

```
[ "/srv/www/config", "/srv/www/shared" ].each do |path|
  directory path do
    mode 0755
    owner 'root'
    group 'root'
    recursive true
    action :create
  end
end
```

Nachfolgend werden Attribute für Unterverzeichnisnamen, Modus, Besitzer und Gruppenwerte definiert.

```
default['createdir']['shared_dir'] = 'shared'
default['createdir']['config_dir'] = 'config'
default['createdir']['mode'] = 0755
default['createdir']['owner'] = 'root'
default['createdir']['group'] = 'root'
```

Beachten Sie Folgendes:

- Jede Definition beginnt mit einem Attributtyp.

Wenn ein Attribut mehr als einmal definiert ist — vielleicht in verschiedenen Attributdateien — gibt der Attributtyp die Priorität des Attributs an, die bestimmt, welche Definition in das Knotenobjekt aufgenommen wird. Weitere Informationen finden Sie unter Priorität von Attributen. Alle Definitionen in diesem Beispiel weisen den Attributtyp `default` auf, der üblicherweise für diesen Zweck verwendet wird.

- Die Attribute haben verschachtelte Namen.

  Das Knotenobjekt ist im Wesentlichen eine Hash-Tabelle, die beliebig tief verschachtelt werden kann. Daher lassen sich auch Attributnamen verschachteln, was gängige Praxis ist. Diese Attributdatei folgt der Standardvorgehensweise und verwendet eine verschachtelte Datei mit dem Rezeptbuch-Namen `createdir` als erstes Element.

Hier wird "createdir" als erstes Element verwendet, weil bei der Chef-Ausführung die Attribute aus allen Rezeptbüchern in das Knotenobjekt eingebunden werden. Bei AWS OpsWorks Stacks enthält das Knotenobjekt zusätzlich zu allen von Ihnen definierten Attributen eine große Anzahl von Attributen aus den integrierten Kochbüchern. Durch das Einbeziehen des Rezeptbuch-Namens in den Attributnamen werden Namenskonflikte mit Attributen aus anderen Rezeptbüchern vermieden. Dies gilt besonders für Attributnamen wie `port` oder `user`. Vergeben Sie keine Attributnamen wie z. B. `[:apache2][:user]`, außer Sie möchten den Attributwert überschreiben. Weitere Informationen finden Sie unter Verwenden von benutzerdefinierten Rezeptbuchattributen.

Im folgenden Beispiel wird das ursprüngliche Rezept mit Attributen anstelle von fest programmierten Werten gezeigt.

```
[ "/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**
>
> Wenn Sie einen Attributwert in eine Zeichenfolge einbinden möchten, umschließen Sie diesen mit `#{}`. Im vorigen Beispiel wird "shared" mit `#{node['createdir']['shared_dir']}` zu "/srv/www/" hinzugefügt.

So führen Sie das Rezept aus

1. Führen Sie `kitchen destroy` aus, damit Sie mit einer neuen Instance beginnen können.
2. Ersetzen Sie den Code in `recipes/default.rb` durch das vorige Rezeptbeispiel.
3. Erstellen Sie für `createdir` das Unterverzeichnis `attributes` und fügen Sie die Datei `default.rb` mit den Attributdefinitionen hinzu.
4. Bearbeiten Sie `.kitchen.yml`, um CentOS aus der Liste der Plattformen zu entfernen.
5. Führen Sie `kitchen converge` aus und melden Sie sich anschließend an der Instance an, um zu prüfen, ob `/srv/www/shared` und `/srv/www/config` vorhanden sind.

> **Note**
>
> Bei AWS OpsWorks Stacks bietet die Definition von Werten als Attribute einen zusätzlichen Vorteil. Sie können benutzerdefiniertes JSON verwenden, um diese Werte pro Stack oder sogar pro Bereitstellung zu überschreiben. Dies kann in vielen Fällen sinnvoll sein, z. B. in den folgenden:
>
> - Sie können das Verhalten Ihrer Rezepte anpassen, wie z. B. die Konfigurationseinstellungen oder Benutzernamen, ohne das Rezeptbuch zu verändern.
>
> Beispielsweise können Sie dasselbe Rezeptbuch für unterschiedliche Stacks einsetzen und die wichtigsten Konfigurationseinstellungen für einen bestimmten Stack mit den benutzerdefinierten JSON-Daten angeben. Auf diese Weise müssen Sie weder die Zeit noch den Aufwand für eine Anpassung des Rezeptbuchs aufbringen noch für jeden Stack ein anderes Rezeptbuch verwenden.
> - Es ist nicht nötig, potenziell vertrauliche Informationen (wie z. B. Datenbank-Passwörter) im Rezeptbuch-Repository zu hinterlegen.
>
> Stattdessen können Sie mittels eines Attributs einen Standardwert festlegen und dann mit den benutzerdefinierten JSON-Daten diesen Wert mit dem echten Wert überschreiben.

> Weitere Informationen zur Verwendung der benutzerdefinierten JSON-Daten zum Überschreiben von Attributen finden Sie unter Überschreiben der Attribute.

Die Attributdatei hat den Namen `default.rb`, da es sich um eine (wenn auch sehr einfache) Ruby-Anwendung handelt. Das heißt, Sie können beispielsweise mithilfe der Bedingungslogik die Attributwerte auf Basis des Betriebssystems angeben. Unter Bedingungslogik haben Sie einen anderen Unterverzeichnisnamen für die verschiedenen Linux-Familien im Rezept angegeben. Wenn Sie eine Attributdatei nutzen, können Sie stattdessen die Bedingungslogik in die Attributdatei einbinden.

In der folgenden Attributdatei wird mit `value_for_platform` ein anderer `['shared_dir']`-Attributwert auf Basis des Betriebssystems angegeben. Für andere Bedingungen können Sie die `if-elsif-else`-Logik von Ruby oder eine `case`-Anweisung verwenden.

```
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'
```

So führen Sie das Rezept aus

1. Führen Sie `kitchen destroy` aus, damit Sie mit einer neuen Instance beginnen können.
2. Ersetzen Sie den Code in `attributes/default.rb` durch das vorherige Beispiel.
3. Bearbeiten Sie `.kitchen.yml` und fügen Sie wie unter Bedingungslogik beschrieben eine CentOS-Plattform zum Abschnitt mit den Plattformen hinzu.
4. Führen Sie `kitchen converge` aus und melden Sie sich anschließend an den Instances an, um zu prüfen, ob die Verzeichnisse vorhanden sind.

Wenn Sie fertig sind, führen Sie `kitchen destroy` aus, um die Instance zu beenden. Im nächsten Beispiel wird ein neues Rezeptbuch verwendet.

Beispiel 6: Erstellen von Dateien

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Nachdem Sie Verzeichnisse erstellt haben, müssen diese meist mit Konfigurationsdateien, Datendateien usw. gefüllt werden. In diesem Thema werden zwei Möglichkeiten vorgestellt, mit denen Sie Dateien auf einer Instance installieren können.

Themen

- Installieren einer Datei mithilfe eines Rezeptbuchs
- Erstellen einer Datei mithilfe einer Vorlage

Installieren einer Datei mithilfe eines Rezeptbuchs

Die einfachste Möglichkeit zum Installieren einer Datei auf einer Instance bietet eine `cookbook_file`-Ressource, mit der eine Datei aus dem Rezeptbuch kopiert und am angegebenen Speicherort auf der Instance eingefügt wird (dies gilt für Linux- und Windows-Systeme). In diesem Beispiel wird das in Beispiel 3: Erstellen von Verzeichnissen verwendete Rezept erweitert, um nach der Verzeichniserstellung eine Datendatei `/srv/www/shared` hinzuzufügen. Hier ist zu Referenzzwecken das ursprüngliche Rezept.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end
```

So richten Sie das Rezeptbuch ein

1. Erstellen Sie ein Verzeichnis in `opsworks_cookbooks` namens `createfile` und öffnen Sie es.
2. Fügen Sie eine Datei `metadata.rb` zu `createfile` mit dem folgenden Inhalt hinzu:

```
name "createfile"
version "0.1.0"
```

3. Initialisieren und konfigurieren Sie Test Kitchen wie unter Beispiel 1: Installieren von Paketen beschrieben und entfernen Sie CentOS aus der Liste `platforms`.
4. Fügen Sie ein Unterverzeichnis `recipes` zu `createfile` hinzu.

Die zu installierende Datei enthält folgende JSON-Daten.

```
{
  "my_name" : "myname",
  "your_name" : "yourname",
  "a_number" : 42,
  "a_boolean" : true
}
```

So richten Sie die Datendatei ein

1. Fügen Sie ein Unterverzeichnis `files` zu `createfile` und ein Unterverzeichnis `default` zu `files` hinzu. Alle Dateien, die Sie mit `cookbook_file` installieren, müssen in einem Unterverzeichnis von `files` sein – in diesem Beispiel in `files/default`.

> **Note**
>
> Falls Sie unterschiedliche Dateien für verschiedene Systeme angeben möchten, können Sie die einzelnen systemspezifischen Dateien in einem Unterordner platzieren, der nach dem jeweiligen System benannt ist (z. B. `files/ubuntu`). Von der `cookbook_file`-Ressource wird dann die geeignete systemspezifische Datei kopiert, sofern vorhanden. Andernfalls wird die Datei `default` verwendet. Weitere Informationen finden Sie unter cookbook_file.

2. Erstellen Sie eine Datei `example_data.json` mit dem JSON-Objekt aus dem vorigen Beispiel und fügen Sie sie zu Verzeichnis `files/default` hinzu.

Mit folgendem Rezept wird `example_data.json` an einen angegebenen Speicherort kopiert.

```
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
```

Nachdem von der Verzeichnisressource das Verzeichnis `/srv/www/shared` erstellt wurde, kopiert die Ressource `cookbook_file` die Datei `example_data.json` in das Verzeichnis und legt zudem den Benutzer, die Gruppe und den Modus für die Datei fest.

> **Note**
>
> Mit der `cookbook_file`-Ressource wird eine neue Aktion eingeführt: `create_if_missing`. Sie könnten auch eine `create`-Aktion verwenden, aber diese würde eine vorhandene Datei überschreiben. Falls nichts überschrieben werden soll, verwenden Sie `create_if_missing`. Damit wird die Datei `example_data.json` nur installiert, wenn sie nicht bereits vorhanden ist.

So führen Sie das Rezept aus

1. Führen Sie `kitchen destroy` aus, damit Sie mit einer neuen Instance beginnen können.
2. Erstellen Sie die Datei `default.rb`, die das vorherige Rezept enthält, und speichern Sie diese in `recipes`.

3. Führen Sie `kitchen converge` aus und melden Sie sich anschließend an der Instance an, um zu prüfen, ob die Datei `/srv/www/shared` in `example_data.json` vorhanden ist.

Erstellen einer Datei mithilfe einer Vorlage

Die `cookbook_file`-Ressource ist für einige Zwecke sehr gut geeignet, jedoch werden von ihr alle im Rezeptbuch vorhandenen Dateien installiert. Eine `template`-Ressource bietet eine flexiblere Methode, um eine Datei auf einem Windows- oder Linux-System zu installieren, da die Datei dynamisch aus einer Vorlage erstellt wird. Somit können Sie die Inhalte der Datei zur Laufzeit bestimmen und bei Bedarf ändern. Beispielsweise können Sie beim Start einer Instance eine bestimmte Einstellung in der Konfigurationsdatei vorgeben und diese später ändern, wenn Sie weitere Instances zum Stack hinzufügen.

Im Beispiel wird das Rezeptbuch `createfile` so angepasst, dass mit einer `template`-Ressource eine leicht abgewandelte Version von `example_data.json` installiert wird.

So sieht die installierte Datei aus.

```
{
  "my_name" : "myname",
  "your_name" : "yourname",
  "a_number" : 42,
  "a_boolean" : true,
  "a_string" : "some string",
  "platform" : "ubuntu"
}
```

"template"-Ressourcen werden in der Regel in Verbindung mit Attributdateien verwendet, daher wird auch in diesem Beispiel eine zur Definition der folgenden Werte genutzt.

```
default['createfile']['my_name'] = 'myname'
default['createfile']['your_name'] = 'yourname'
default['createfile']['install_file'] = true
```

So richten Sie das Rezeptbuch ein

1. Löschen Sie das Verzeichnis `createfile` und dessen Inhalte aus dem Rezeptbuch `files`.

2. Fügen Sie das Unterverzeichnis `attributes` zu `createfile` hinzu. Fügen Sie außerdem die Datei `default.rb` mit den vorherigen Attributdefinitionen zu `attributes` hinzu.

Bei einer Vorlage handelt es sich um eine `.erb`-Datei, die im Grunde genommen eine Kopie der letzten Datei ist, in der einige Inhalte durch Platzhalter dargestellt werden. Bei der Dateierstellung mithilfe der `template`-Ressource werden die Vorlageninhalte in die angegebene Datei kopiert und die Platzhalter mit den zugewiesenen Werten überschrieben. Hier ist die Vorlage für `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'] %>"
}
```

Die Werte `<%=...%>` sind die Platzhalter.

- `<%=node[...]%>` stellt den Wert eines Knotenattributs dar.

  In diesem Beispiel ist der Wert "your_name" ein Platzhalter für einen Attributwert aus der Attributdatei des Rezeptbuchs.
- `<%=@...%>` stellt den Wert einer Variable dar, die in der "template"-Ressource definiert ist (wird später erläutert).

So erstellen Sie die Vorlagendatei

1. Fügen Sie ein Unterverzeichnis `templates` zu dem Rezeptbuch `createfile` und ein Unterverzeichnis `default` zu `templates` hinzu.

   > **Note**
   >
   > Das Verzeichnis `templates` funktioniert auf die gleiche Weise wie das Verzeichnis `files`. Sie können systemspezifische Vorlagen in einem Unterverzeichnis, das z. B. wie `ubuntu` nach dem jeweiligen System benannt ist, speichern. Von der `template`-

Ressource wird dann die geeignete systemspezifische Datei genutzt, sofern vorhanden. Andernfalls wird die Vorlage `default` verwendet.

2. Erstellen Sie eine Datei namens `example_data.json.erb` und legen Sie sie in das Verzeichnis `templates/default`. Der Vorlagenname ist beliebig wählbar, wird aber durch die Dateinamenerweiterung `.erb` (einschließlich anderer Erweiterungen) gekennzeichnet.

Das folgende Rezept verwendet eine `template`-Ressource, um `/srv/www/shared/example_data.json` zu erstellen.

```
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
```

Von der `template`-Ressource wird mithilfe einer Vorlage die Datei `example_data.json` erstellt und in `/srv/www/shared` installiert.

- Der Vorlagenname `/srv/www/shared/example_data.json` gibt den Pfad und den Namen der installierten Datei an.
- Das `source`-Attribut gibt die Vorlage an, mit der die Datei erstellt wurde.
- Das `mode`-Attribut gibt den Modus der installierten Datei an.
- Die Ressource definiert die beiden Variablen `a_boolean_var` und `a_string_var`.

Wenn die Datei `example_data.json` von der Ressource erstellt wird, werden die Variablenplatzhalter in der Vorlage mit den entsprechenden Werten aus der Ressource überschrieben.

- Mit dem `only_if` -Wächterattribut wird die Ressource angewiesen, die Datei nur zu erstellen, sofern `['createfile']['install_file']` auf den Wert `true` gesetzt ist.

So führen Sie das Rezept aus

1. Führen Sie `kitchen destroy` aus, damit Sie mit einer neuen Instance beginnen können.
2. Ersetzen Sie den Code in `recipes/default.rb` durch das vorherige Beispiel.
3. Führen Sie `kitchen converge` aus und melden Sie sich anschließend an der Instance an, um zu prüfen, ob die Datei in `/srv/www/shared` mit dem korrekten Inhalt vorhanden ist.

Wenn Sie fertig sind, führen Sie `kitchen destroy` aus und fahren damit die Instance herunter. Im nächsten Abschnitt wird ein neues Rezeptbuch verwendet.

Beispiel 7: Ausführen von Befehlen und Skripts

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Chef-Ressourcen können zahlreiche Aufgaben auf Instances ausführen, aber gelegentlich ist es sinnvoller, einen Shell-Befehl oder ein Skript zu verwenden. Das ist beispielsweise der Fall, wenn Sie bereits Skripts für bestimmte Aufgaben einsetzen. Dann ist es einfacher, diese weiterzuverwenden, anstatt neuen Code zu implementieren. In diesem Abschnitt erfahren Sie, wie Befehle oder Skripts auf einer Instance ausgeführt werden.

Themen

- Ausführen von Befehlen
- Ausführen von Skripts

Ausführen von Befehlen

Die `script`-Ressource kann einen oder mehrere Befehle ausführen. Sie unterstützt die Befehlsinterpreter csh, bash, Perl, Python und Ruby, sodass sie sowohl auf Linux- als auch auf Windows-Systemen eingesetzt werden kann (sofern die entsprechenden Befehlsinterpreter installiert sind). In diesem Thema wird gezeigt, wie ein einfacher bash-Befehl auf einer Linux-Instance ausgeführt wird. Chef unterstützt zudem powershell_script und batch-Ressourcen für die Skriptausführung auf Windows. Weitere Informationen finden Sie unter Ein PowerShell Windows-Skript ausführen.

Dies sind Ihre ersten Schritte

1. Erstellen Sie ein Verzeichnis in `opsworks_cookbooks` namens `script` und öffnen Sie es.
2. Fügen Sie eine Datei `metadata.rb` zu `script` mit dem folgenden Inhalt hinzu:

   ```
   name "script"
   version "0.1.0"
   ```

3. Initialisieren und konfigurieren Sie Test Kitchen wie unter Beispiel 1: Installieren von Paketen beschrieben und entfernen Sie CentOS aus der Liste `platforms`.
4. Erstellen Sie in `script` ein Verzeichnis namens `recipes`.

Sie können Befehle direkt mit der `script`-Ressource ausführen, aber Chef unterstützt auch eine Reihe von Ressourcenversionen, die auf bestimmte Befehlsinterpreter ausgerichtet und nach diesen benannt sind. Im folgenden Rezept wird ein einfaches bash-Skript mithilfe einer `bash`-Ressource ausgeführt.

```
bash "install_something" do
  user "root"
  cwd "/tmp"
  code <<-EOH
    touch somefile
  EOH
  not_if do
    File.exists?("/tmp/somefile")
  end
end
```

Die `bash`-Ressource ist wie folgt konfiguriert.

- Sie verwendet die `run`-Standardaktion zur Ausführung der Befehle im `code`-Block.

  In diesem Beispiel ist nur ein einziger Befehl (`touch somefile`) vorhanden, aber ein `code`-Block kann mehrere Befehle enthalten.
- Das `user`-Attribut gibt den Benutzer an, der den Befehl ausführt.
- Das `cwd`-Attribut gibt das Arbeitsverzeichnis an.

  In diesem Beispiel wird von `touch` eine Datei im Verzeichnis `/tmp` erstellt.
- Das `not_if`-Wächterattribut weist die Ressource an, keine Aktion auszuführen, wenn die Datei bereits vorhanden ist.

So führen Sie das Rezept aus

1. Erstellen Sie die Datei `default.rb`, die den vorherigen Beispiel-Code enthält, und speichern Sie diese in `recipes`.
2. Führen Sie `kitchen converge` aus und melden Sie sich anschließend an der Instance an, um zu prüfen, ob die Datei in `/tmp` vorhanden ist.

Ausführen von Skripts

Die `script`-Ressource ist sehr praktisch, insbesondere, wenn Sie nur einen oder zwei Befehle ausführen möchten. In vielen Fällen ist es jedoch sinnvoller, das Skript in einer Datei zu speichern und diese auszuführen. Mit der `execute`-Ressource wird eine angegebene ausführbare Datei, einschließlich Skriptdateien, auf Linux- oder Windows-Systemen ausgeführt. In diesem Thema wird das Rezeptbuch `script` aus dem vorherigen Beispiel angepasst, um mithilfe von `execute` ein einfaches Shell-Skript auszuführen. Sie können das Beispiel problemlos für komplexere Skripte oder andere ausführbare Dateitypen erweitern.

So richten Sie die Skriptdatei ein

1. Fügen Sie ein Unterverzeichnis `files` zu `script` und ein Unterverzeichnis `default` zu `files` hinzu.
2. Erstellen Sie eine Datei namens `touchfile`, die Folgendes enthält, und fügen Sie sie zu `files/default` hinzu. In diesem Beispiel wird ein gängiger bash-Interpreter genutzt, aber Sie können bei Bedarf auch einen Interpreter wählen, der für Ihre Shell-Umgebung geeignet ist.

```
#!/usr/bin/env bash
touch somefile
```

Die Skriptdatei kann beliebig viele Befehle enthalten. In diesem Beispielskript ist aus Gründen der Übersichtlichkeit nur ein `touch`-Befehl enthalten.

Mit dem folgenden Rezept wird das Skript ausgeführt.

```
cookbook_file "/tmp/touchfile" do
  source "touchfile"
  mode 0755
end

execute "touchfile" do
  user "root"
  cwd "/tmp"
  command "./touchfile"
end
```

Von der Ressource `cookbook_file` wird die Skriptdatei in `/tmp` kopiert, zudem wird der Modus festgelegt, damit die Datei ausführbar ist. Die `execute`-Ressource führt dann die Datei wie folgt aus:

- Das `user`-Attribut gibt den Benutzer des Befehls an (in diesem Beispiel `root`).
- Das Attribut `cwd` gibt das Arbeitsverzeichnis an (in diesem Beispiel `/tmp`).
- Das `command`-Attribut gibt das auszuführende Skript an (in diesem Beispiel `touchfile`), das im Arbeitsverzeichnis gespeichert ist.

So führen Sie das Rezept aus

1. Ersetzen Sie den Code in `recipes/default.rb` durch das vorherige Beispiel.
2. Führen Sie `kitchen converge` aus und melden Sie sich anschließend an der Instance an, um zu prüfen, ob `/tmp` nun die Skriptdatei, den Modus 0755 und `somefile` enthält.

Wenn Sie fertig sind, führen Sie `kitchen destroy` aus und fahren damit die Instance herunter. Im nächsten Abschnitt wird ein neues Rezeptbuch verwendet.

Beispiel 8: Verwalten von Services

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Paketen wie z. B. Anwendungsservern ist in der Regel ein Service zugeordnet, der gestartet, gestoppt, neu gestartet usw. werden muss. Beispielsweise müssen Sie den Tomcat-Service nach der Paketinstallation und nach beendetem Instance-Start starten sowie nach jeder Änderung der Konfigurationsdatei neu starten. In diesem Thema werden die Grundlagen für die Verwaltung eines Service auf einer Linux-Instance am Beispiel eines Tomcat-Anwendungsservers dargelegt. Die "service"-Ressource funktioniert auf Windows-Instances auf dieselbe Weise, allerdings gibt es ein paar kleine Unterschiede. Weitere Informationen finden Sie unter `service`.

> **Note**
>
> Im Beispiel wird eine sehr minimale Tomcat-Installation ausgeführt. Sie reicht aus, um die Grundlagen für die Verwendung einer `service`-Ressource zu veranschaulichen. Ein Beispiel für die Rezeptimplementierung auf einem Tomcat-Server mit mehr Funktionen finden Sie unter Erstellen eines benutzerdefinierten Tomcat-Server-Layers.

Themen

Definieren und Starten eines Service

In diesem Abschnitt werden die Grundlagen zum Definieren und Starten eines Service erläutert.

Dies sind Ihre ersten Schritte

1. Erstellen Sie ein Verzeichnis im Verzeichnis `opsworks_cookbooks` namens `tomcat` und öffnen Sie es.
2. Fügen Sie eine Datei `metadata.rb` zu `tomcat` mit dem folgenden Inhalt hinzu:

   ```
   name "tomcat"
   version "0.1.0"
   ```
3. Initialisieren und konfigurieren Sie Test Kitchen wie unter Beispiel 1: Installieren von Paketen beschrieben und entfernen Sie CentOS aus der Liste `platforms`.
4. Fügen Sie ein Unterverzeichnis `recipes` zu `tomcat` hinzu.

Verwenden Sie eine `service`-Ressource für die Serviceverwaltung. Mit dem folgenden Standardrezept wird Tomcat installiert und der Service gestartet.

```
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
```

Vom Rezept werden folgende Schritte ausgeführt:

- Die `execute`-Ressource führt `apt-get update` aus, um aktuelle Systemupdates zu installieren.

  Für die in diesem Beispiel verwendete Ubuntu-Instanz müssen Sie die Updates installieren, bevor Sie Tomcat installieren. Bei anderen Systemen können die Anforderungen abweichen.
- Die `package`-Ressource installiert Tomcat 7.
- Das enthaltene `tomcat::service`-Rezept definiert den Service (wird später erläutert).

- Die `service`-Ressource startet den Tomcat-Service.

  Mit dieser Ressource können Sie auch andere Befehle ausgeben, z. B. den Service stoppen und neu starten.

Im folgenden Beispiel finden Sie das `tomcat::service`-Rezept.

```
service 'tomcat' do
  service_name "tomcat7"
  supports :restart => true, :reload => false, :status => true
  action :nothing
end
```

Mit diesem Rezept wird die Tomcat-Servicedefinition wie folgt erstellt:

- Der Ressourcenname `tomcat` wird von anderen Rezepten als Referenz auf den Service verwendet.

  Beispielsweise wird `default.rb` von `tomcat` für den Servicestart referenziert.
- Die `service_name`-Ressource gibt den Servicenamen an.

  Wenn Sie die Services auf der Instance auflisten, wird der Tomcat-Service mit dem Namen "tomcat7" angegeben.
- `supports` gibt an, wie Chef die Befehle `restart`, `reload` und `status` des Service verwaltet.
  - Der Wert `true` gibt an, dass Chef das "init"-Skript oder einen anderen Serviceanbieter zum Ausführen des Befehls verwenden kann.
  - `false` gibt an, dass Chef versuchen muss, den Befehl anderweitig auszuführen.

Beachten Sie, dass `action` auf `:nothing` festgelegt ist. Damit wird die Ressource angewiesen, keine Aktion auszuführen. Die "service"-Ressource unterstützt Aktionen wie `start` und `restart`. Dieses Rezeptbuch folgt jedoch der Standardvorgehensweise und verwendet eine Servicedefinition, bei der keine Aktion erfolgt. Der Service wird anderweitig gestartet bzw. neu gestartet. Jedes Rezept, über das ein Service gestartet oder neu gestartet wird, muss diesen zunächst definieren. Die einfachste Methode ist daher, die Servicedefinition in einem separaten Rezept zu speichern und dieses bei Bedarf in andere Rezepte einzubinden.

> **Note**
>
> Der Einfachheit halber wird im Standardrezept dieses Beispiels eine `service`-Ressource verwendet, um den Service nach Ausführung der Servicedefinition zu starten. In einer Produktionsimplementierung wird ein Service in der Regel mit `notifies` gestartet oder neu gestartet (wird später erläutert).

So führen Sie das Rezept aus

1. Erstellen Sie die Datei `default.rb`, die das Standardrezeptbeispiel enthält, und speichern Sie diese in `recipes`.
2. Erstellen Sie die Datei `service.rb`, die das Servicedefinitionsbeispiel enthält, und speichern Sie diese in `recipes`.
3. Führen Sie `kitchen converge` aus, melden Sie sich anschließend an der Instance an und führen Sie den folgenden Befehl aus, um zu prüfen, ob der Service ausgeführt wird.

   ```
   sudo service tomcat7 status
   ```

> **Note**
>
> Falls Sie `service.rb` getrennt von `default.rb` ausführen möchten, müssen Sie `.kitchen.yml` ändern und `tomcat::service` zur Ausführungsliste hinzufügen. Wenn Sie ein Rezept einbinden, wird dessen Code vor der Rezeptausführung jedoch in das übergeordnete Rezept übernommen. Daher ist `service.rb` im Grunde genommen ein Teil von `default.rb` und erfordert keinen eigenen Eintrag in der Ausführungsliste.

Verwenden von "notifies" für den Start oder Neustart eines Service

In einer Produktionsimplementierung wird ein Service in der Regel nicht mit `service` gestartet oder neu gestartet. Stattdessen wird `notifies` zu verschiedenen Ressourcen hinzugefügt. Wenn Sie beispielsweise den Service nach einer Änderung der Konfigurationsdatei neu starten möchten, binden Sie `notifies` in die zugehörige `template`-Ressource ein. Die Verwendung von `notifies` bietet im Vergleich zur `service`-Ressource für den expliziten Neustart des Service die folgenden Vorteile.

- Das `notifies`-Element startet den Service nur dann neu, wenn die zugehörige Konfigurationsdatei geändert wurde. Das Risiko eines unnötigen Serviceneustarts fällt damit weg.
- Chef startet den Service höchstens einmal am Ende jeder Ausführung neu, unabhängig von der `notifies`-Anzahl pro Ausführung.

  Beispielsweise können in der Chef-Ausführung mehrere "template"-Ressourcen enthalten sein, von denen jede eine andere Konfigurationsdatei ändert und nach der Dateiänderung einen Neustart des Service erfordert. Sie möchten aber in der Regel den Service nur einmal neu starten, und zwar am Ende der Chef-Ausführung. Andernfalls wird möglicherweise ein Service neu gestartet, der nach dem vorherigen Neustart noch nicht wieder betriebsbereit ist, und das könnte zu Fehlern führen.

In diesem Beispiel wird `tomcat::default` angepasst, um eine `template`-Ressource einzubinden, die den Service mithilfe von `notifies` neu startet. Für ein realistisches Beispiel würden Sie eine "template"-Ressource nutzen, die eine benutzerdefinierte Version von einer Tomcat-Konfigurationsdatei erstellt, aber diese sind meist sehr lang und komplex. Der Einfachheit halber wird in diesem Beispiel die "template"-Ressource aus Erstellen einer Datei mithilfe einer Vorlage verwendet. Sie hat keinerlei Verbindung zu Tomcat, bietet aber eine einfache Möglichkeit, die Verwendung von `notifies` darzustellen. Ein Beispiel für die Vorlagenverwendung beim Erstellen von Tomcat-Konfigurationsdateien finden Sie unter Einrichtungsrezepte.

So richten Sie das Rezeptbuch ein

1. Fügen Sie ein Unterverzeichnis `templates` zu `tomcat` und ein Unterverzeichnis `default` zu `templates` hinzu.
2. Kopieren Sie die Vorlage `example_data.json.erb` aus dem Rezeptbuch `createfile` in das Verzeichnis `templates/default`.
3. Fügen Sie ein Unterverzeichnis `attributes` zu `tomcat` hinzu.
4. Kopieren Sie die Attributdatei `default.rb` aus dem Rezeptbuch `createfile` in das Verzeichnis `attributes`.

Im folgenden Rezept wird `notifies` für den Neustart des Tomcat-Service verwendet.

```
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
```

Im Beispiel wird das Rezept aus Erstellen einer Datei mithilfe einer Vorlage mit dem Rezept aus dem vorherigen Abschnitt zusammengeführt. Dabei gibt es zwei wichtige Änderungen:

- Die `service`-Ressource ist nach wie vor vorhanden, erfüllt aber nun einen anderen Zweck.

  Die `:enable`-Aktion aktiviert den Tomcat-Service beim Start.
- In die "template"-Ressource ist `notifies` eingebunden, sodass der Tomcat-Service bei einer Änderung der Datei `example_data.json` neu gestartet wird.

  Auf diese Weise wird sichergestellt, dass der Service bei der Tomcat-Installation gestartet und nach jeder Konfigurationsänderung neu gestartet wird.

So führen Sie das Rezept aus

1. Führen Sie `kitchen destroy` aus, damit Sie mit einer neuen Instance beginnen können.
2. Ersetzen Sie den Code in `default.rb` durch das vorherige Beispiel.
3. Führen Sie `kitchen converge` aus und melden Sie sich anschließend an der Instance an, um zu prüfen, ob der Service ausgeführt wird.

> **Note**
>
> Wenn Sie einen Service neu starten möchten, aber das Rezept keine Ressource wie z. B. `template` enthält, die `notifies` unterstützt, können Sie stattdessen eine `execute`-Dummy-Ressource nutzen. Beispiel
>
> ```
> execute 'trigger tomcat service restart' do
>   command 'bin/true'
>   notifies :restart, resources(:service => 'tomcat')
> end
> ```
>
> Die `execute`-Ressource muss ein `command`-Attribut aufweisen, auch wenn Sie die Ressource nur zur Ausführung von `notifies` einsetzen. In diesem Beispiel wird diese Anforderung durch die Ausführung von `/bin/true` umgangen. Dieser Shell-Befehl gibt einfach einen Erfolgscode zurück.

Beispiel 9: Verwenden von Amazon EC2 EC2-Instances

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Bis zu diesem Zeitpunkt haben Sie Instanzen lokal in ausgeführt. VirtualBox Dies ist zwar schnell und einfach, aber Sie werden Ihre Rezepte irgendwann auf einer Amazon EC2 EC2-Instance

testen wollen. Insbesondere, wenn Sie Rezepte auf Amazon Linux ausführen möchten, ist es nur auf Amazon EC2 verfügbar. Sie können ein ähnliches System wie CentOS für die vorläufige Implementierung und das Testen verwenden, aber die einzige Möglichkeit, Ihre Rezepte auf Amazon Linux vollständig zu testen, ist eine Amazon EC2 EC2-Instance.

In diesem Thema wird gezeigt, wie Rezepte auf einer Amazon EC2 EC2-Instance ausgeführt werden. Dafür verwenden Sie Test Kitchen und Vagrant auf dieselbe Weise wie in den vorherigen Abschnitten, allerdings gibt es zwei Unterschiede:

- Anstelle von Vagrant wird `kitchen-ec2` als Treiber eingesetzt.
- Die `.kitchen.yml` Datei des Kochbuchs muss mit den Informationen konfiguriert werden, die zum Starten der Amazon EC2 EC2-Instance erforderlich sind.

> **Note**
>
> Alternativ können Sie das Vagrant-Plug-in `vagrant-aws` verwenden. Weitere Informationen finden Sie unter Vagrant AWS Provider.

Sie benötigen AWS-Anmeldeinformationen, um eine Amazon EC2 EC2-Instance zu erstellen. Falls Sie noch kein AWS-Konto haben, können Sie wie folgt eines anlegen.

Melden Sie sich für eine an AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu https://aws.amazon.com/ auf und klicken Sie auf Mein Konto.

Erstellen Sie einen Benutzer mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen Sie einen Benutzer mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis im AWS IAM Identity Center Benutzerhandbuch.

Melden Sie sich als Benutzer mit Administratorzugriff an

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal.

Weisen Sie weiteren Benutzern Zugriff zu

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

   Anweisungen finden Sie im Benutzerhandbuch unter Einen Berechtigungssatz erstellen.AWS IAM Identity Center
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Anweisungen finden Sie im AWS IAM Identity Center Benutzerhandbuch unter Gruppen hinzufügen.

Sie sollten einen IAM-Benutzer mit Zugriffsberechtigungen für Amazon EC2 erstellen und den Zugriff und die geheimen Schlüssel des Benutzers an einem sicheren Ort auf Ihrer Workstation speichern. Test Kitchen verwendet diese Anmeldeinformationen für die Instance-Erstellung. Am besten stellen Sie die Anmeldeinformationen für Test Kitchen bereit, indem Sie die Schlüssel den folgenden Umgebungsvariablen auf der Workstation zuweisen.

> **Warning**
>
> IAM-Benutzer verfügen über langfristige Anmeldeinformationen, was ein Sicherheitsrisiko darstellt. Um dieses Risiko zu minimieren, empfehlen wir, diesen Benutzern nur die Berechtigungen zu gewähren, die sie für die Ausführung der Aufgabe benötigen, und diese Benutzer zu entfernen, wenn sie nicht mehr benötigt werden.

- AWS_ACCESS_KEY — der Zugriffsschlüssel Ihres Benutzers, der ungefähr so aussehen wird wie AKIAIOSFODNN7EXAMPLE.

- AWS_SECRET_KEY — der geheime Schlüssel Ihres Benutzers, der etwa wie wjalrxUTNFEMI/k7mdeng/CYEXAMPLEKEY aussehen wird. bPxRfi

Auf diese Weise wird vermieden, dass Ihre Kontodaten versehentlich offengelegt werden, wenn Sie z. B. ein Projekt, das Ihre Anmeldeinformationen enthält, in ein öffentliches Repository hochladen.

So richten Sie das Rezeptbuch ein

1. Für die Verwendung des `kitchen-ec2`-Treibers muss das Paket `ruby-dev` auf Ihrem System installiert sein. Im folgenden Beispiel wird veranschaulicht, wie Sie das Paket mit `aptitude` auf einem Ubuntu-System installieren.

   ```
   sudo aptitude install ruby1.9.1-dev
   ```

2. Bei `kitchen-ec2` handelt es sich um einen Gem-Treiber, der wie folgt installiert wird:

   ```
   gem install kitchen-ec2
   ```

   Abhängig von Ihrer Workstation benötigen Sie hierfür womöglich `sudo` oder einen Ruby-Umgebungsmanager wie RVM. Dieses Verfahren wurde mit Version 0.8.0 des `kitchen-ec2`-Treibers getestet, jedoch gibt es inzwischen neuere Versionen. Um eine bestimmte Version zu installieren, führen Sie `gem install kitchen-ec2 -v <version number>` aus.

3. Sie müssen ein Amazon EC2 SSH-Schlüsselpaar angeben, das Test Kitchen verwenden kann, um eine Verbindung mit der Instance herzustellen. Wenn Sie kein Amazon EC2 EC2-Schlüsselpaar haben, finden Sie unter Amazon EC2 EC2-Schlüsselpaare Informationen darüber, wie Sie eines erstellen. Das Schlüsselpaar muss sich in derselben AWS-Region befinden wie die Instance. Das Beispiel verwendet US West (Nordkalifornien).

   Wenn Sie ein Schlüsselpaar ausgewählt haben, erstellen Sie in `opsworks_cookbooks` das Unterverzeichnis `ec2_keys` und kopieren die Datei mit dem privaten Schlüssel des Schlüsselpaars (`.pem`) in das Unterverzeichnis. Der private Schlüssel wird nur in `ec2_keys` gespeichert, um den Code ein wenig zu vereinfachen; er kann überall auf dem System gespeichert werden.

4. Erstellen Sie ein Unterverzeichnis von `opsworks_cookbooks` namens `createdir-ec2` und öffnen Sie es.

5. Fügen Sie eine Datei `metadata.rb` zu `createdir-ec2` mit dem folgenden Inhalt hinzu:

```
name "createdir-ec2"
version "0.1.0"
```

6. Initialisieren Sie Test Kitchen wie unter Beispiel 1: Installieren von Paketen beschrieben. Im folgenden Abschnitt wird die Konfiguration beschrieben `.kitchen.yml`, die für Amazon EC2 EC2-Instances erheblich komplizierter ist.
7. Fügen Sie ein Unterverzeichnis `recipes` zu `createdir-ec2` hinzu.

.kitchen.yml für Amazon EC2 konfigurieren

Sie konfigurieren `.kitchen.yml` mit den Informationen, die der `kitchen-ec2` Treiber benötigt, um eine entsprechend konfigurierte Amazon EC2 EC2-Instance zu starten. Im Folgenden finden Sie ein Beispiel für eine `.kitchen.yml` Datei für eine Amazon Linux-Instance in der Region USA West (Nordkalifornien).

```
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:
```

In den Abschnitten `provisioner` und `suites` können Sie die Standardeinstellungen verwenden, aber für `driver` und `platforms` müssen diese angepasst werden. In diesem Beispiel werden nur die minimal erforderlichen Einstellungen angepasst, ansonsten werden Standardwerte genutzt. Eine vollständige Liste der `kitchen-ec2`-Einstellungen finden Sie unter Kitchen::Ec2: A Test Kitchen Driver for Amazon EC2.

Im Beispiel werden die folgenden `driver`-Attribute festgelegt. Es wird vorausgesetzt, dass Sie den Zugriffsschlüssel und den geheimen Schlüssel Ihres Benutzers den Standardumgebungsvariablen zugewiesen haben (wie zuvor erläutert). Diese Schlüssel werden vom Treiber standardmäßig verwendet. Andernfalls müssen Sie die Schlüssel explizit deklarieren, indem Sie `aws_access_key_id` und `aws_secret_access_key` zu den `driver`-Attributen hinzufügen und auf die entsprechenden Schlüsselwerte festlegen.

Name

(Erforderlich) Dieses Attribut muss auf `ec2` festgelegt werden.

aws_ssh_key_id

(Erforderlich) Der Name des Amazon EC2 EC2-SSH-Schlüsselpaars, der `US-East1` in diesem Beispiel benannt ist.

transport.ssh_key

(Erforderlich) Die Datei mit dem privaten Schlüssel (`.pem`) zum Schlüssel, den Sie für `aws_ssh_key_id` angegeben haben. In diesem Beispiel heißt die Datei `US-East1.pem` und ist im Verzeichnis `../opsworks/ec2_keys` gespeichert.

Region

(Erforderlich) Die AWS-Region der Instance. In dem Beispiel wird US West (Nordkalifornien) verwendet, was durch`us-west-1`) dargestellt wird.

availability_zone

(Optional) Die Availability Zone (AZ) der Instance. Wenn Sie diese Einstellung weglassen, verwendet Test Kitchen eine standardmäßige Availability Zone für die angegebene Region, die `us-west-1b` für USA West (Nordkalifornien) gilt. Möglicherweise ist diese Standard-AZ für Ihr Konto nicht verfügbar. In dem Fall müssen Sie explizit eine Availability Zone angeben. Das für

diese Beispiele verwendete Konto unterstützt `us-west-1b` nicht, daher wird `us-west-1c` im Beispiel explizit angegeben.

require_chef_omnibus

Mit dem Wert `true` stellt diese Einstellung sicher, dass das Omnibus-Installationsprogramm für die Installation von `chef-client` auf allen Plattform-Instances verwendet wird.

security_group_ids

(Optional) Eine Liste mit Sicherheitsgruppen-IDs, die für die Instance gelten. Mit dieser Einstellung wird die Sicherheitsgruppe `default` für die Instance verwendet. Stellen Sie sicher, dass die für den Dateneingang festgelegten Regeln der Sicherheitsgruppe eingehende SSH-Verbindungen zulassen. Andernfalls kann Test Kitchen nicht mit der Instance kommunizieren. Wenn Sie die Sicherheitsgruppe `default` nutzen, müssen Sie diese möglicherweise entsprechend anpassen. Weitere Informationen finden Sie unter Amazon EC2-Sicherheitsgruppen.

subnet_id

Die ID des Ziel-Subnetzes für die Instance (falls zutreffend).

associate_public_ip

Sie können Amazon EC2 der Instance eine öffentliche IP-Adresse zuordnen lassen, wenn Sie über das Internet auf die Instance zugreifen möchten.

interface

Der Konfigurationstyp des Host-Namens, der für den Zugriff auf die Instance verwendet wird. Gültige Werte sind `dns`, `public`, `private` oder `private_dns`. Falls Sie keinen Wert für dieses Attribut angeben, wird die Host-Namenskonfiguration von `kitchen-ec2` in folgender Reihenfolge eingerichtet. Wenn Sie dieses Attribut weglassen, wird kein Konfigurationstyp festgelegt.

1. DNS-Name
2. Öffentliche IP-Adresse
3. Private IP-Adresse
4. Private DNS name (Privater DNS-Name)

> **Important**
>
> Anstatt Ihre Kontoanmeldedaten für den Zugriff und die geheimen Schlüssel zu verwenden, sollten Sie einen Benutzer erstellen und diese Anmeldeinformationen an Test Kitchen

weitergeben. Weitere Informationen finden Sie unter Bewährte Methoden für die Verwaltung von AWS-Zugriffsschlüsseln.
Achte darauf, es nicht an einem öffentlich zugänglichen Ort zu speichern, z. B. wenn du es `.kitchen.yml` in ein öffentliches Repository GitHub oder ein Bitbucket-Repository hochlädst. Dadurch könnten Ihre Anmeldeinformationen offengelegt und die Sicherheit Ihres Kontos beeinträchtigt werden.

Der `kitchen-ec2`-Treiber unterstützt standardmäßig die folgenden Plattformen:

- 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

Wenn Sie eine oder mehrere dieser Plattformen verwenden möchten, fügen Sie zu `platforms` die entsprechenden Plattformnamen hinzu. Der `kitchen-ec2`-Treiber wählt automatisch ein geeignetes AMI aus und generiert einen SSH-Benutzernamen. Sie können andere Plattformen verwenden — in diesem Beispiel wird Amazon Linux verwendet —, aber Sie müssen die folgenden Attribute explizit angeben. `platforms`

Name

Der Name der Plattform. In diesem Beispiel wird Amazon Linux verwendet, folglich ist `name` auf `amazon` gesetzt.

driver

Die `driver`-Attribute, zu denen die nachfolgenden zählen:

- `image_id`— Das AMI der Plattform, das zur angegebenen Region gehören muss. Das Beispiel verwendet `ami-ed8e9284` ein Amazon Linux-AMI aus der Region USA West (Nordkalifornien).

- `transport.username`— Der SSH-Benutzername, den Test Kitchen für die Kommunikation mit der Instance verwenden wird.

  Verwenden Sie `ec2-user` für Amazon Linux. Für andere AMIs werden ggf. andere Benutzernamen herangezogen.

Ersetzen Sie den Code in `.kitchen.yml` durch das Beispiel und weisen Sie den kontobezogenen Attributen (wie `aws_access_key_id`) entsprechende Werte zu.

Ausführen des Rezepts

Im Beispiel wird das Rezept aus Iteration verwendet.

So führen Sie das Rezept aus

1. Erstellen Sie die Datei `default.rb` mit folgendem Code und speichern Sie diese im Rezeptbuch-Ordner `recipes`.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end
```

2. Führen Sie `kitchen converge` aus, um das Rezept auszuführen. Beachten Sie, dass die Ausführung dieses Befehls aufgrund der Zeit, die zum Starten und Initialisieren einer Amazon EC2 EC2-Instance benötigt wird, länger dauert als bei den vorherigen Beispielen.
3. Gehen Sie zur Amazon EC2 EC2-Konsole, wählen Sie die Region USA West (Nordkalifornien) aus und klicken Sie im Navigationsbereich auf Instances. Die neu erstellte Instance wird in der Liste angezeigt.
4. Führen Sie `kitchen login` den Befehl aus, um sich bei der Instance anzumelden, genau wie Sie es bei Instances getan haben, die in VirtualBox ausgeführt werden. Die neu erstellten Verzeichnisse werden unter `/srv` angezeigt. Für die Verbindung zur Instance können Sie auch Ihren bevorzugten SSH-Client nutzen.

Nächste Schritte

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Kapitel haben Sie die Grundlagen der Implementierung von Chef-Rezeptbüchern kennengelernt, aber das ist noch lange nicht alles:

- In den Beispielen wurde veranschaulicht, wie Sie einige der gängigen Ressourcen verwenden, aber es gibt noch zahlreiche mehr.

  Für die vorgestellten Ressourcen wurden in den Beispielen nur ein paar der verfügbaren Attribute und Aktionen genutzt. Ausführlichere Informationen finden Sie unter About Resources and Providers.
- In den Beispielen wurden nur die Core-Elemente eines Rezeptbuchs angewendet: `recipes`, `attributes`, `files` und `templates`.

  Rezeptbücher können noch zahlreiche andere Elemente enthalten, z. B. `libraries`, `definitions` und `specs`. Weitere Informationen finden Sie unter Chef documentation.
- In den Beispielen wurde Test Kitchen nur als gute Möglichkeit genutzt, um Instances zu starten, Rezepte auszuführen und sich an der Instance anzumelden.

  In erster Linie ist Test Kitchen aber eine Testplattform, mit der Sie viele Tests für Ihre Rezepte ausführen können. Lesen Sie die Test Kitchen-Anleitung und lernen Sie die Testfunktionen kennen (sofern noch nicht geschehen).
- Implementierung von Kochbüchern für Stacks AWS OpsWorksbietet einige fortgeschrittenere Beispiele und zeigt, wie Kochbücher für Stacks implementiert werden. AWS OpsWorks

Implementierung von Kochbüchern für Stacks AWS OpsWorks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In Rezeptbücher – Grundlagen haben sie Rezeptbücher und Rezepte kennengelernt. Die Beispiele in diesem Abschnitt waren vom Design her einfach und funktionieren auf jeder Instanz, die Chef unterstützt, einschließlich AWS OpsWorks Stacks-Instanzen. Um anspruchsvollere Kochbücher für AWS OpsWorks Stacks zu implementieren, müssen Sie in der Regel alle Vorteile der AWS OpsWorks Stacks-Umgebung nutzen, die sich in vielerlei Hinsicht vom Standard-Chef unterscheidet.

In diesem Thema werden die Grundlagen der Implementierung von Rezepten für AWS OpsWorks Stacks-Instanzen beschrieben.

> **Note**
>
> Wenn Sie mit der Implementierung von Rezeptbüchern noch nicht vertraut sind, arbeiten Sie zunächst Rezeptbücher – Grundlagen durch.

Themen

- Überschreiben von integrierten Vorlagen

Ein Rezept auf einer AWS OpsWorks Stacks-Linux-Instance ausführen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Test Kitchen und Vagrant bieten eine einfache und effiziente Möglichkeit, Kochbücher zu implementieren. Um jedoch zu überprüfen, ob die Rezepte eines Kochbuchs in der Produktion korrekt ausgeführt werden, müssen Sie sie auf einer Stacks-Instanz ausführen. AWS OpsWorks In diesem Thema wird beschrieben, wie Sie ein benutzerdefiniertes Rezeptbuch auf einer AWS OpsWorks Stacks-Linux-Instance installieren und ein einfaches Rezept ausführen. Außerdem finden Sie hier einige Tipps zur effizienten Behebung von Fehlern in Rezepten.

Eine Anleitung zum Ausführen von Rezepten auf Windows-Instances finden Sie unter Ausführen eines Rezepts auf einer Windows-Instance.

Themen

- Erstellen und Ausführen von Rezepten
- Automatisches Ausführen des Rezepts
- Fehlersuche und Fehlerbehebung bei Rezepten

Erstellen und Ausführen von Rezepten

Zunächst müssen Sie einen Stack erstellen. Nachfolgend wird kurz beschrieben, wie Sie für dieses Beispiel einen Stack erstellen. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

So erstellen Sie einen -Stack

1. Öffnen Sie die AWS OpsWorks Stacks-Konsole und klicken Sie auf Add Stack (Stack hinzufügen).

2. Legen Sie die folgenden Einstellungen fest, übernehmen Sie für die restlichen Einstellungen die Standardwerte und klicken Sie auf Add Stack (Stack hinzufügen).

   - Name — OpsTest
   - Standard-SSH-Schlüssel — Ein Amazon EC2 EC2-Schlüsselpaar

   Wenn Sie ein Amazon EC2 EC2-Schlüsselpaar erstellen müssen, finden Sie weitere Informationen unter Amazon EC2 EC2-Schlüsselpaare. Das Schlüsselpaar muss sich in derselben AWS-Region befinden wie die Instance. Das Beispiel verwendet die Standardregion USA West (Oregon).
3. Klicken Sie auf Add a layer (Layer hinzufügen) und fügen Sie dem Stack einen benutzerdefinierten Layer mit folgenden Einstellungen hinzu.

   - Name — OpsTest
   - Kurzname — opstest

   Für Linux-Stacks können Sie einen beliebigen Layer-Typ verwenden. In diesem Beispiel werden jedoch keine der durch die anderen Layer-Typen installierten Pakete benötigt, daher ist es am einfachsten, einen benutzerdefinierten Layer zu verwenden.
4. Fügen Sie dem Layer eine 24/7-Instance mit den Standardeinstellungen hinzu und starten Sie sie.

Während die Instanz gestartet wird — das dauert in der Regel mehrere Minuten — können Sie das Kochbuch erstellen. In diesem Beispiel verwenden wir eine geringfügig angepasste Version des Rezepts aus Bedingungslogik, um ein Datenverzeichnis anzulegen, dessen Name abhängig von der Plattform ist.

So richten Sie das Rezeptbuch ein

1. Erstellen Sie ein Verzeichnis in `opsworks_cookbooks` namens `opstest` und öffnen Sie es.
2. Erstellen Sie eine Datei `metadata.rb` mit dem folgenden Inhalt und speichern Sie sie unter `opstest`.

```
name "opstest"
version "0.1.0"
```

3. Erstellen Sie ein Verzeichnis `recipes` in `opstest`.
4. Erstellen Sie eine Datei `default.rb` mit dem folgenden Rezept und speichern Sie sie im Verzeichnis `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
```

Das Rezept erstellt eine Nachricht durch Aufrufen von `Chef::Log.info`. Sie verwenden Test Kitchen für dieses Beispiel nicht, daher ist die `log` Methode nicht sehr nützlich. `Chef::Log.info`fügt die Nachricht in das Chef-Protokoll ein, das Sie nach Abschluss des Chef-Laufs lesen können. AWS OpsWorks Stacks bietet eine einfache Möglichkeit, diese Protokolle anzuzeigen, wie später beschrieben.

> **Note**
>
> Chef-Protokolle enthalten normalerweise zahlreiche Routinedaten und vergleichsweise uninteressante Informationen. Über das Zeichen "*", das zur Strukturierung der Nachricht verwendet wird, können Sie diese leichter finden.

5. Erstellen Sie ein `.zip`-Archiv von `opsworks_cookbooks`. Um Ihr Kochbuch auf einer AWS OpsWorks Stacks-Instanz zu installieren, müssen Sie es in einem Repository speichern und AWS OpsWorks Stacks die Informationen zur Verfügung stellen, die zum Herunterladen des Kochbuchs auf die Instanz erforderlich sind. Sie können Rezeptbücher in verschiedenen unterstützten Repository-Typen speichern. In diesem Beispiel wird eine Archivdatei mit

den Kochbüchern in einem Amazon S3 S3-Bucket gespeichert. Weitere Informationen zu Rezeptbuch-Repositorys finden Sie unter Rezeptbuch-Repositorys.

> **Note**
>
> Der Einfachheit halber wird in diesem Beispiel das gesamte Verzeichnis `opsworks_cookbooks` archiviert. Dies bedeutet jedoch, dass AWS OpsWorks Stacks alle Kochbücher in `opsworks_cookbooks` die Instanz herunterlädt, obwohl Sie nur eines davon verwenden werden. Um nur das Beispielrezeptbuch zu installieren, erstellen Sie ein neues Verzeichnis und verschieben Sie `opstest` in dieses Verzeichnis. Erstellen Sie dann ein `.zip`-Archiv des übergeordneten Verzeichnisses und verwenden Sie dieses anstelle von `opsworks_cookbooks.zip`.
> Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

6. Laden Sie das Archiv in einen Amazon S3 S3-Bucket hoch, machen Sie das Archiv öffentlich und notieren Sie die URL des Archivs.

Jetzt können Sie das Rezeptbuch installieren und das Rezept ausführen.

So führen Sie das Rezept aus

1. Bearbeiten Sie den Stack, um benutzerdefinierte Rezeptbücher zu aktivieren, und legen Sie folgende Einstellungen fest:

   - Repository-Typ — S3-Archiv
   - Repository-URL — Die URL des Kochbuch-Archivs, die Sie zuvor aufgezeichnet haben

   Verwenden Sie für die übrigen Einstellungen die Standardwerte und klicken Sie auf Save (Speichern), um die Stack-Konfiguration zu aktualisieren und zu speichern.
2. Führen Sie den Stack-Befehl „Update Custom Cookbooks” aus, um die aktuelle Version Ihrer benutzerdefinierten Rezeptbücher auf den Stack-Instances zu installieren. Wenn bereits eine ältere Version der Rezeptbücher installiert ist, werden diese überschrieben.
3. Führen Sie das Rezept aus, indem Sie den Stack-Befehl Execute Recipes ausführen. Achten Sie darauf, dass bei Recipes to execute **`opstest::default`** eingestellt ist. Durch diesen Befehl wird Chef mit der Option `opstest::default` ausgeführt.

Nachdem das Rezept erfolgreich ausgeführt wurde, können Sie es überprüfen.

So überprüfen Sie opstest

1. Werfen Sie zunächst einen Blick in das Chef-Protokoll. Klicken Sie auf show in der Spalte Log der Instance „opstest1”, um das Protokoll anzuzeigen. Blättern Sie nach unten, wo Sie Ihre Protokollmeldung finden.

   ```
   ...
   [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. Melden Sie sich über SSH bei der Instance an und rufen Sie den Inhalt des Verzeichnisses `/srv/www/` auf.

Wenn Sie alle Schritte befolgt haben, werden Sie `/srv/www/config` anstelle des erwarteten Verzeichnisses `/srv/www/shared` sehen. Die folgenden Abschnitte enthalten einige Tipps, um solche Probleme schnell zu beheben.

Automatisches Ausführen des Rezepts

Mit dem Befehl Execute Recipes (Rezepte ausführen) können Sie benutzerdefinierte Rezepte einfach testen, daher wird er auch in den meisten dieser Beispiele verwendet. In der Praxis führen Sie Rezepte jedoch in der Regel zu Standardzeitpunkten im Lebenszyklus einer Instanz aus, z. B. nachdem der Start der Instanz abgeschlossen ist oder wenn Sie eine App bereitstellen. AWS OpsWorks Stacks vereinfacht die Ausführung von Rezepten auf Ihrer Instance, indem es eine Reihe von Lebenszyklusereignissen für jede Ebene unterstützt: Setup, Configure, Deploy, Undeploy und Shutdown. Sie können AWS OpsWorks Stacks veranlassen, ein Rezept automatisch auf den Instanzen einer Ebene auszuführen, indem Sie das Rezept dem entsprechenden Lebenszyklusereignis zuweisen.

Normalerweise erstellen Sie Verzeichnisse, sobald die Instance hochgefahren wurde, also während des Einrichtens. Nachfolgend wird beschrieben, wie Sie das Beispielrezept während des Einrichtens

auf demselben Stack ausführen, den Sie zuvor in diesem Beispiel erstellt haben. Für die anderen Ereignisse können Sie ebenso vorgehen.

So führen Sie Rezepte automatisch beim Einrichten aus

1. Wählen Sie im Navigationsbereich Ebenen aus und klicken Sie dann auf das Stiftsymbol neben dem Link Rezepte der OpsTest Ebene.
2. Fügen Sie **`opstest::default`** zu den Setup-Rezepten des Layers hinzu, klicken Sie auf +, um es dem Layer hinzuzufügen, und wählen Sie Save aus, um die Konfiguration zu speichern.
3. Wählen Sie Instances aus, fügen Sie dem Layer eine weitere Instance hinzu und starten Sie sie.

   Die Instance sollte den Namen `opstest2` haben. Nach Abschluss des Startvorgangs wird AWS OpsWorks Stacks ausgeführt. `opstest::default`
4. Nachdem die Instance `opstest2` online ist, überprüfen Sie, ob das Verzeichnis `/srv/www/shared` angelegt wurde.

> **Note**
>
> Falls Sie den Ereignissen Einrichtung, Konfiguration oder Bereitstellung Rezepte zugewiesen haben, können Sie diese auch mit einem Stack-Befehl (Einrichtung und Konfiguration) oder einem Bereitstellungsbefehl (Bereitstellung) manuell ausführen, um das Ereignis auszulösen. Falls einem Ereignis mehrere Rezepte zugewiesen sind, werden mit diesen Befehlen alle Rezepte eines Ereignisses ausgeführt.

Fehlersuche und Fehlerbehebung bei Rezepten

Falls Sie nicht die erwarteten Ergebnisse erhalten oder Ihre Rezepte gar nicht erst erfolgreich ausgeführt werden, beginnt die Fehlersuche normalerweise im Chef-Protokoll. Es enthält eine detaillierte Beschreibung der Rezeptausführung sowie interne Protokollmeldungen Ihrer Rezepte. Die Protokolle sind insbesondere bei fehlgeschlagenen Rezepten hilfreich, da in diesem Fall sowohl der Fehler als auch ein Stacktrace im Chef-Protokoll gespeichert werden.

Wenn das Rezept erfolgreich ausgeführt wurde wie in diesem Beispiel, ist das Chef-Protokoll allerdings oft keine große Hilfe. In diesem Fall sollten Sie sich einfach das Rezept und insbesondere die ersten Zeilen genauer ansehen:

```
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 ist ein angemessener Ersatz für Amazon Linux, wenn Sie Rezepte auf Vagrant testen. Jetzt führen Sie Rezepte jedoch auf einer tatsächlichen Amazon Linux-Instance aus. Der Plattformwert für Amazon Linux ist `amazon`. Dieser Wert ist im Aufruf `value_for_platform` nicht enthalten, daher erstellt das Rezept standardmäßig das Verzeichnis `/srv/www/config`. Weitere Informationen zur Fehlerbehebung finden Sie unter Handbuch zur Fehlersuche und Fehlerbehebung.

Nachdem Sie nun das Problem gefunden haben, können Sie das Rezept entsprechend aktualisieren und noch einmal testen. Sie könnten zu den ursprünglichen Quelldateien zurückkehren, aktualisieren`default.rb`, ein neues Archiv auf Amazon S3 hochladen und so weiter. Dies ist jedoch ziemlich mühsam und zeitaufwändig. Nachfolgend wird ein wesentlich schnellerer Ansatz vorgestellt, der insbesondere bei einfachen Fehlern wie in unserem Beispiel hilfreich ist: Wir bearbeiten das Rezept auf der Instance.

So bearbeiten Sie ein Rezept auf einer Instance

1. Melden Sie sich über SSH bei der Instance an und führen Sie `sudo su` aus, um Administratorberechtigungen zu erhalten. Sie benötigen diese Root-Berechtigungen, um auf das Rezeptbuch-Verzeichnis zugreifen zu können.
2. AWS OpsWorks Stacks speichert Ihr Kochbuch in`/opt/aws/opsworks/current/site-cookbooks`, navigieren Sie also zu. `/opt/aws/opsworks/current/site-cookbooks/opstest/recipes`

> **Note**
>
> AWS OpsWorks Stacks speichert auch eine Kopie Ihrer Kochbücher in. `/opt/aws/opsworks/current/merged-cookbooks` Bearbeiten Sie dieses Rezeptbuch nicht. Wenn Sie das Rezept ausführen, kopiert AWS OpsWorks Stacks das Kochbuch von `.../site-cookbooks` nach`.../merged-cookbooks`, sodass alle Änderungen, die Sie daran vornehmen, überschrieben werden. `.../merged-cookbooks`

3. Bearbeiten Sie die Datei `default.rb` mit einem Texteditor auf der Instance und ersetzen Sie `centos` durch `amazon`. Ihr Rezept sollte jetzt wie folgt aussehen.

```
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"
)
...
```

Führen Sie den Stack-Befehl Execute Recipe (Rezept ausführen) erneut aus, um zu überprüfen, ob der Fehler behoben wurde. Die Instance sollte jetzt ein Verzeichnis `/srv/www/shared` haben. Wenn Sie weitere Änderungen an einem Rezept vornehmen möchten, können Sie den Befehl Execute Recipe jederzeit erneut ausführen. Die Instance muss dafür nicht jedes Mal angehalten und neu gestartet werden. Wenn Sie mit dem Rezept zufrieden sind, vergessen Sie nicht, den Code auch in Ihrem Quellrezeptbuch zu aktualisieren.

> **Note**
>
> Wenn Sie Ihr Rezept einem Lebenszyklusereignis zugewiesen haben, sodass AWS OpsWorks Stacks es automatisch ausführt, können Sie Execute Recipe jederzeit verwenden, um das Rezept erneut auszuführen. Sie können das Rezept auch beliebig oft erneut ausführen, ohne die Instanz neu zu starten, indem Sie die AWS OpsWorks Stacks-Konsole verwenden, um das entsprechende Ereignis manuell auszulösen. Auf diese Weise werden jedoch alle Rezepte des Ereignisses ausgeführt. Zur Erinnerung:
>
> - Verwenden Sie einen Stack-Befehl, um Einrichtungs- oder Konfigurationsereignisse auszulösen.
> - Verwenden Sie einen Bereitstellungsbefehl, um Bereitstellungsereignisse oder Ereignisse zum Aufheben der Bereitstellung auszulösen.

Ausführen eines Rezepts auf einer Windows-Instance

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Dieses Thema ist im Grunde eine verkürzte Version von Ausführen eines Rezepts auf einer Linux-Instance und erläutert, wie Sie ein Rezept auf einem Windows-Stack ausführen. Wir empfehlen Ihnen daher, zunächst Ausführen eines Rezepts auf einer Linux-Instance durchzuarbeiten, da Sie dort detailliertere Informationen finden, die für beide Betriebssysteme relevant sind.

Eine Beschreibung der Ausführung von Rezepten auf AWS OpsWorks Stacks-Linux-Instances finden Sie unter. Ausführen eines Rezepts auf einer Linux-Instance

Themen

- Aktivieren von RDP-Zugriff
- Erstellen und Ausführen von Rezepten
- Automatisches Ausführen des Rezepts

Aktivieren von RDP-Zugriff

Falls Sie dies noch nicht getan haben, müssen Sie zunächst eine Sicherheitsgruppe mit einer Regel für eingehenden Datenverkehr einrichten, die RDP-Zugriff für Ihre Instances zulässt. Sie benötigen diese Gruppe beim Erstellen des Stacks.

Wenn Sie den ersten Stack in einer Region erstellen, erstellt AWS OpsWorks Stacks eine Reihe von Sicherheitsgruppen. Dazu gehört eine mit dem Namen etwa`AWS-OpsWorks-RDP-Server`, die AWS OpsWorks Stacks an alle Windows-Instanzen anhängt, um RDP-Zugriff zu ermöglichen. Standardmäßig sind in diesen Sicherheitsgruppe jedoch keine Regeln enthalten. Daher müssen Sie eine Regel für den eingehenden Datenverkehr zum Zulassen von RDP-Zugriff auf Ihre Instances hinzufügen.

So ermöglichen Sie den RDP-Zugriff

1. Öffnen Sie die Amazon EC2 EC2-Konsole, stellen Sie sie auf die Region des Stacks ein und wählen Sie im Navigationsbereich Sicherheitsgruppen aus.
2. Wählen Sie AWS- OpsWorks -RDP-Server, klicken Sie auf die Registerkarte Inbound und dann auf Bearbeiten.
3. Fügen Sie eine Regel mit folgenden Einstellungen hinzu:

   - Typ — RDP
   - Quelle — Die zulässigen Quell-IP-Adressen.

     In der Regel erlauben Sie eingehende RDP-Anfragen von Ihrer eigenen IP-Adresse oder einem festen IP-Adressbereich (üblicherweise der IP-Adressbereich Ihres Unternehmens).

> **Note**
>
> Wie nachfolgend beschrieben müssen Sie auch die Benutzerberechtigungen bearbeiten, um RDP-Zugriff für reguläre Benutzer zu ermöglichen.

Weitere Informationen finden Sie unter Anmelden mit RDP.

Erstellen und Ausführen von Rezepten

Nachfolgend wird kurz beschrieben, wie Sie für dieses Beispiel einen Stack erstellen. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

Erstellen eines Stacks

1. Öffnen Sie die AWS OpsWorks Stacks-Konsole und wählen Sie Add Stack (Stack hinzufügen) aus. Legen Sie die folgenden Einstellungen fest, übernehmen Sie für die restlichen Einstellungen die Standardwerte und wählen Sie Add Stack (Stack hinzufügen) aus.

   - Name — WindowsRecipeTest
   - Region — USA West (Oregon)

     Dieses Beispiel funktioniert in jeder Region, wir empfehlen jedoch, US West (Oregon) für Tutorials zu verwenden.
   - Standardbetriebssystem — Microsoft Windows Server 2012 R2

2. Wählen Sie Add a layer (Layer hinzufügen) aus und fügen Sie dem Stack einen benutzerdefinierten Layer mit folgenden Einstellungen hinzu:

   - Name — RecipeTest
   - Kurzname — Recipetest

3. Fügen Sie dem RecipeTest Layer eine 24/7-Instanz mit Standardeinstellungen hinzu und starten Sie ihn.

   AWS OpsWorks Stacks weist diese Instanz automatisch `AWS-OpsWorks-RDP-Server` zu, sodass sich autorisierte Benutzer bei der Instanz anmelden können.

4. Wählen Sie Permissions (Berechtigungen), dann Edit (Bearbeiten) und anschließend SSH/RDP und sudo/admin aus. Reguläre Benutzer benötigen zusätzlich zur Sicherheitsgruppe `AWS-OpsWorks-RDP-Server` diese Autorisierung, um sich bei der Instance anzumelden.

   > **Note**
   >
   > Sie können sich auch als Administrator anmelden, allerdings mit einem anderen Verfahren. Weitere Informationen finden Sie unter Anmelden mit RDP.

Während die Instanz gestartet wird — das dauert in der Regel mehrere Minuten — können Sie das Kochbuch erstellen. Über das Rezept in diesem Beispiel, bei dem es sich um eine für Windows angepasste Version des Rezepts aus Beispiel 3: Erstellen von Verzeichnissen handelt, wird ein Datenverzeichnis erstellt.

> **Note**
>
> Bei der Implementierung von Kochbüchern für AWS OpsWorks Stacks-Windows-Instances verwenden Sie eine etwas andere Verzeichnisstruktur als bei der Implementierung von Cookbooks für Stacks-Linux-Instances. AWS OpsWorks Weitere Informationen finden Sie unter Rezeptbuch-Repositorys.

So richten Sie das Rezeptbuch ein

1. Erstellen Sie ein Verzeichnis `windowstest` und öffnen Sie es.
2. Erstellen Sie eine Datei `metadata.rb` mit dem folgenden Inhalt und speichern Sie sie unter `windowstest`.

```
name "windowstest"
version "0.1.0"
```

3. Erstellen Sie ein Verzeichnis `recipes` in `windowstest`.
4. Erstellen Sie eine Datei `default.rb` mit dem folgenden Rezept und speichern Sie sie im Verzeichnis `recipes`.

```
Chef::Log.info("******Creating a data directory.******")

directory 'C:\data' do
  rights :full_control, 'instance_name\username'
  inherits false
  action :create
end
```

   Ersetzen Sie *username* durch Ihren Benutzernamen.
5. Speichern Sie das Rezeptbuch in einem Repository.

   Um Ihr Kochbuch auf einer AWS OpsWorks Stacks-Instanz zu installieren, müssen Sie es in einem Repository speichern und Stacks die Informationen zur Verfügung stellen, die zum AWS OpsWorks Herunterladen des Kochbuchs auf die Instanz erforderlich sind. Sie können Windows-Rezeptbücher als Archivdatei in einem S3-Bucket oder in einem Git-Repository speichern. In diesem Beispiel verwenden wir einen S3-Bucket, daher müssen Sie ein ZIP-Archiv des Verzeichnisses `windowstest` erstellen. Weitere Informationen zu Rezeptbuch-Repositorys finden Sie unter Rezeptbuch-Repositorys.
6. Laden Sie das Archiv in einen S3-Bucket hoch, veröffentlichen Sie das Archiv und notieren Sie sich die URL des Archivs. Sie können auch ein privates Archiv verwenden. Für dieses Beispiel ist ein öffentliches Archiv jedoch ausreichend und einfacher zu handhaben.

   Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Jetzt können Sie das Rezeptbuch installieren und das Rezept ausführen.

So führen Sie das Rezept aus

1. Bearbeiten Sie den Stack, um benutzerdefinierte Rezeptbücher zu aktivieren, und legen Sie folgende Einstellungen fest:

   - Repository-Typ — S3-Archiv
   - Repository-URL — Die URL des Kochbuch-Archivs, die Sie zuvor aufgezeichnet haben

   Übernehmen Sie für die übrigen Einstellungen die Standardwerte und wählen Sie Save aus, um die Stack-Konfiguration zu aktualisieren und zu speichern.
2. Führen Sie den Stack-Befehl „Update Custom Cookbooks" aus, um die aktuelle Version Ihrer benutzerdefinierten Rezeptbücher auf den Stack-Instances einschließlich Online-Instances zu installieren. Wenn bereits eine ältere Version der Rezeptbücher installiert ist, werden diese überschrieben.
3. Nachdem die benutzerdefinierten Rezeptbücher aktualisiert wurden, führen Sie das Rezept mithilfe des Stack-Befehls Execute Recipes aus. Achten Sie darauf, dass bei Recipes to execute **`windowstest::default`** eingestellt ist. Durch diesen Befehl wird Chef mit Ihrem Rezept ausgeführt.

Nachdem das Rezept erfolgreich ausgeführt wurde, können Sie es überprüfen.

So überprüfen Sie windowstest

1. Sehen Sie sich das Chef-Protokoll an. Wählen Sie show in der Spalte Log der Instance „opstest1" aus, um das Protokoll anzuzeigen. Blättern Sie nach unten, wo Sie Ihre Protokollmeldung finden.

```
...
[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. Wählen Sie Instances und anschließend rdp in der Spalte Actions (Aktionen) der Instance aus und fordern Sie ein RDP-Passwort mit einer angemessenen Ablaufzeit an. Kopieren Sie den DNS-Namen, den Benutzernamen und das Passwort. Sie können sich mithilfe eines RDP-Clients wie dem Windows Remote Desktop Connection-Client mit diesen Informationen bei der Instance anmelden und überprüfen, ob das Verzeichnis `c:\data` angelegt wurde. Weitere Informationen finden Sie unter Anmelden mit RDP.

> **Note**
>
> Wenn Ihr Rezept nicht ordnungsgemäß ausgeführt wurde, finden Sie unter Fehlersuche und Fehlerbehebung bei Rezepten Tipps zur Fehlerbehebung. Die meisten dieser Tipps lassen sich auch auf Windows-Instances übertragen. Wenn du deine Lösung testen möchtest, indem du das Rezept auf der Instanz bearbeitest, suche in dem `C:\chef\cookbooks` Verzeichnis, in dem AWS OpsWorks Stacks benutzerdefinierte Kochbücher installiert, nach deinem Kochbuch.

Automatisches Ausführen des Rezepts

Mit dem Befehl Execute Recipes (Rezepte ausführen) können Sie benutzerdefinierte Rezepte einfach testen, daher wird er auch in den meisten dieser Beispiele verwendet. In der Praxis führen Sie Rezepte jedoch in der Regel zu Standardzeitpunkten im Lebenszyklus einer Instanz aus, z. B. nachdem die Instanz den Startvorgang abgeschlossen hat oder wenn Sie eine App bereitstellen. AWS OpsWorks Stacks vereinfacht die Ausführung von Rezepten auf Ihrer Instance, indem es eine Reihe von Lebenszyklusereignissen für jede Ebene unterstützt: Setup, Configure, Deploy, Undeploy und Shutdown. Sie können AWS OpsWorks Stacks veranlassen, ein Rezept automatisch auf den Instanzen einer Ebene auszuführen, indem Sie das Rezept dem entsprechenden Lebenszyklusereignis zuweisen.

Normalerweise erstellen Sie Verzeichnisse, sobald die Instance hochgefahren wurde, also während des Einrichtens. Nachfolgend wird beschrieben, wie Sie das Beispielrezept während des Einrichtens auf demselben Stack ausführen, den Sie zuvor in diesem Beispiel erstellt haben. Für die anderen Ereignisse können Sie ebenso vorgehen.

So führen Sie Rezepte automatisch beim Einrichten aus

1. Wählen Sie im Navigationsbereich „Ebenen“ und dann das Stiftsymbol neben dem Link „Rezepte“ der RecipeTest Ebene aus.

2. Fügen Sie **`windowstest::default`** zu den Setup-Rezepten des Layers hinzu, klicken Sie auf **+**, um es dem Layer hinzuzufügen, und wählen Sie Save aus, um die Konfiguration zu speichern.
3. Wählen Sie Instances aus, fügen Sie dem Layer eine weitere Instance hinzu und starten Sie sie.

   Die Instance sollte den Namen `recipetest2` haben. Nach Abschluss des Startvorgangs wird AWS OpsWorks Stacks ausgeführt. `windowstest::default`
4. Nachdem die Instance `recipetest2` online ist, überprüfen Sie, ob das Verzeichnis `c:\data` angelegt wurde.

> **Note**
>
> Falls Sie den Ereignissen Einrichtung, Konfiguration oder Bereitstellung Rezepte zugewiesen haben, können Sie diese auch mit einem Stack-Befehl (Einrichtung und Konfiguration) oder einem Bereitstellungsbefehl (Bereitstellung) auch manuell ausführen, um das Ereignis auszulösen. Falls einem Ereignis mehrere Rezepte zugewiesen sind, werden mit diesen Befehlen alle Rezepte eines Ereignisses ausgeführt.

Ein Windows-Skript ausführen PowerShell

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> In diesen Beispielen wird davon ausgegangen, dass Sie das Beispiel Ausführen eines Rezepts auf einer Windows-Instance bereits durchgearbeitet haben. Falls Sie das noch nicht getan haben, holen Sie das nun nach. Insbesondere wird darin beschrieben, wie Sie für Ihre Instances RDP-Zugriff aktivieren.

Eine Möglichkeit, ein Rezept Aufgaben auf einer Windows-Instanz ausführen zu lassen — insbesondere Aufgaben, für die es keine entsprechende Chef-Ressource gibt — besteht darin, das Rezept ein Windows-Skript ausführen zu lassen. PowerShell In diesem Abschnitt werden Sie in die Grundlagen eingeführt, indem beschrieben wird, wie Sie ein PowerShell Windows-Skript verwenden, um eine Windows-Funktion zu installieren.

Die `powershell_script`Ressource führt PowerShell Windows-Cmdlets auf einer Instanz aus. Im folgenden Beispiel wird ein WindowsFeature Install-Cmdlet verwendet, um einen XPS-Viewer auf der Instanz zu installieren.

Nachfolgend wird kurz beschrieben, wie Sie für dieses Beispiel einen Stack erstellen. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

Erstellen eines Stacks

1. Öffnen Sie die AWS OpsWorks Stacks-Konsole und wählen Sie Add Stack (Stack hinzufügen) aus. Legen Sie die folgenden Einstellungen fest, übernehmen Sie für die restlichen Einstellungen die Standardwerte und klicken Sie auf Add Stack (Stack hinzufügen).

   - Name — PowerShellTest
   - Region — USA West (Oregon)

     Dieses Beispiel funktioniert in jeder Region, wir empfehlen jedoch, US West (Oregon) für Tutorials zu verwenden.
   - Standardbetriebssystem — Microsoft Windows Server 2012 R2
2. Wählen Sie Add a layer (Layer hinzufügen) aus und fügen Sie dem Stack einen benutzerdefinierten Layer mit folgenden Einstellungen hinzu:

   - Name — PowerShell
   - Kurzname — Powershell
3. Fügen Sie dem PowerShell Layer eine 24/7-Instanz mit Standardeinstellungen hinzu und starten Sie ihn.
4. Wählen Sie Permissions (Berechtigungen), dann Edit (Bearbeiten) und anschließend SSH/RDP und sudo/admin aus. Als regulärer Benutzer benötigen Sie zusätzlich zur Sicherheitsgruppe `AWS-OpsWorks-RDP-Server` diese Autorisierung, um sich bei der Instance anzumelden.

Während die Instanz gestartet wird — das dauert in der Regel mehrere Minuten — können Sie das Kochbuch erstellen. Über das Rezept in diesem Beispiel, bei dem es sich um eine für Windows

angepasste Version des Rezepts aus Beispiel 3: Erstellen von Verzeichnissen handelt, wird ein Datenverzeichnis erstellt.

So richten Sie das Rezeptbuch ein

1. Erstellen Sie ein Verzeichnis `powershell` und öffnen Sie es.
2. Erstellen Sie eine Datei `metadata.rb` mit dem folgenden Inhalt und speichern Sie sie unter `windowstest`.

```
name "powershell"
version "0.1.0"
```

3. Erstellen Sie ein Verzeichnis `recipes` innerhalb des Verzeichnisses `powershell`.
4. Erstellen Sie eine Datei `default.rb` mit dem folgenden Rezept und speichern Sie sie im Verzeichnis `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
```

   - Mithilfe der Ressource `powershell_script` wird ein Cmdlet ausgeführt, um den XPS-Viewer zu installieren.

     In diesem Beispiel wird nur ein Cmdlet ausgeführt. Der `code`-Block kann jedoch beliebig viele Befehlszeilen enthalten.
   - Das `guard_interpreter` Attribut weist Chef an, die 64-Bit-Version von Windows zu verwenden. PowerShell
   - Über das Wächterattribut `not_if` wird sichergestellt, dass Chef die Funktion nur dann installiert, wenn sie nicht bereits installiert ist.
5. Erstellen Sie ein `.zip`-Archiv des Verzeichnisses `powershell`.

6. Laden Sie das Archiv in einen Amazon S3 S3-Bucket hoch, machen Sie das Archiv öffentlich und notieren Sie die URL des Archivs. Sie können auch ein privates Archiv verwenden. Für dieses Beispiel ist ein öffentliches Archiv jedoch ausreichend und einfacher zu handhaben.

   Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Jetzt können Sie das Rezeptbuch installieren und das Rezept ausführen.

So führen Sie das Rezept aus

1. Bearbeiten Sie den Stack, um benutzerdefinierte Rezeptbücher zu aktivieren, und legen Sie folgende Einstellungen fest:

   - Repository-Typ — S3-Archiv
   - Repository-URL — Die URL des Kochbuch-Archivs, die Sie zuvor aufgezeichnet haben

   Übernehmen Sie für die übrigen Einstellungen die Standardwerte und wählen Sie Save aus, um die Stack-Konfiguration zu aktualisieren und zu speichern.
2. Führen Sie den Stack-Befehl Update Custom Cookbooks aus, um die aktuelle Version Ihrer benutzerdefinierten Rezeptbücher auf der Instance zu installieren.
3. Nachdem die benutzerdefinierten Rezeptbücher über den Befehl Update Custom Cookbooks aktualisiert wurden, führen Sie das Rezept mithilfe des Stack-Befehls Execute Recipes aus. Achten Sie darauf, das bei Recipes to execute **`powershell::default`** eingestellt ist.

> **Note**
>
> In diesem Beispiel wird der Einfachheit halber Execute Recipes verwendet, aber normalerweise lassen Sie AWS OpsWorks Stacks Ihre Rezepte automatisch ausführen, indem Sie sie dem entsprechenden Lebenszyklusereignis zuweisen. Sie können diese Rezepte auch durch manuelles Auslösen des Ereignisses ausführen. Verwenden Sie für Einrichtungs- und Konfigurationsereignisse einen Stack-Befehl und für Bereitstellungsereignisse und für Ereignisse zum Aufheben der Bereitstellung einen Bereitstellungsbefehl.

Nachdem das Rezept erfolgreich ausgeführt wurde, können Sie es überprüfen.

So überprüfen Sie das powershell-Rezept

1. Sehen Sie sich das Chef-Protokoll an. Klicken Sie auf show in der Spalte Log der Instance „powershell1”, um das Protokoll anzuzeigen. Blättern Sie nach unten, wo Sie Ihre Protokollmeldung finden.

```
...
[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. Melden Sie sich über RDP bei der Instance an und öffnen Sie das Menü Start. XPS Viewer sollte unter Windows Zubehör aufgelistet sein.

Nachahmen der Stack-Konfiguration und Bereitstellungsattribute auf Vagrant

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Dieses Thema bezieht sich nur auf Linux-Instances. Test Kitchen unterstützt Windows noch nicht, daher werden Sie alle Windows-Beispiele auf AWS OpsWorks Stacks-Instanzen ausführen.

AWS OpsWorks Stacks fügt dem Knotenobjekt für jede Instanz in Ihrem Stack für jedes Lebenszyklusereignis Stackkonfigurations- und Bereitstellungsattribute hinzu. Diese Attribute sind ein Snapshot der Stack-Konfiguration einschließlich der Konfiguration der einzelnen Layer und deren Online-Instances, der Konfiguration der einzelnen bereitgestellten Apps usw. Da sich diese Attribute im Node-Objekt befinden, kann auf sie mit jedem Rezept zugegriffen werden. Die meisten Rezepte für AWS OpsWorks Stacks-Instances verwenden eines oder mehrere dieser Attribute.

Eine Instanz, die in einer Vagrant-Box ausgeführt wird, wird nicht von AWS OpsWorks Stacks verwaltet, sodass ihr Knotenobjekt standardmäßig keine Stackkonfigurations- und Bereitstellungsattribute enthält. Sie können jedoch der Test Kitchen-Umgebung entsprechend geeignete Attribute hinzufügen. Test Kitchen fügt dann die Attribute zum Knotenobjekt der Instanz hinzu, und Ihre Rezepte können auf die Attribute zugreifen, genauso wie sie es auf einer AWS OpsWorks Stacks-Instanz tun würden.

In diesem Thema wird gezeigt, wie Sie eine Kopie von geeigneten Stack-Konfigurations- und Bereitstellungsattributen erstellen, die Attribute auf einer Instance installieren und dann darauf zugreifen.

> **Note**
>
> Wenn Sie Ihre Rezepte mit Test Kitchen testen, können Sie die Stack-Konfiguration und Bereitstellungs-JSON auch mit fauxhai simulieren.

So richten Sie das Rezeptbuch ein

1. Erstellen Sie ein Unterverzeichnis von `opsworks_cookbooks` namens `printjson` und öffnen Sie es.
2. Initialisieren und konfigurieren Sie Test Kitchen wie unter Beispiel 1: Installieren von Paketen beschrieben.
3. Fügen Sie zwei Unterverzeichnisse zu `printjson` hinzu: `recipes` und `environments`.

Sie können die Stack-Konfigurations- und Bereitstellungsattribute beispielsweise dadurch nachahmen, dass Sie Ihrem Rezeptbuch eine Attributdatei mit den entsprechenden Definitionen hinzufügen. Besser ist es jedoch, dafür die Test Kitchen-Umgebung zu nutzen. Hierfür gibt es zwei grundlegende Ansätze:

- Fügen Sie Attributdefinitionen zu `.kitchen.yml` hinzu.

  Dieser Ansatz ist insbesondere bei einer geringen Anzahl an Attributen hilfreich. Weitere Informationen finden Sie unter kitchen.yml.

- Definieren Sie die Attribute in einer Umgebungsdatei und verweisen Sie in `.kitchen.yml` auf diese Datei.

  Dieser Ansatz ist für Stack-Konfigurations- und Bereitstellungsattribute in der Regel besser, da die Umgebungsdatei bereits im JSON-Format vorliegt. Sie können eine Kopie der Attribute im JSON-Format von einer geeigneten AWS OpsWorks Stacks-Instanz abrufen und sie einfach einfügen. In allen Beispielen wird eine solche Umgebungsdatei verwendet.

Am einfachsten erstellen Sie Stack-Konfigurations- und Bereitstellungsattribute für Ihr Rezeptbuch, indem Sie einen entsprechend konfigurierten Stack erstellen und die entsprechenden Attribute aus einer Instance im JSON-Format kopieren. Damit Ihre Test Kitchen-Umgebungsdatei übersichtlich bleibt, können Sie diese JSON-Datei anschließend bearbeiten und alle Attribute löschen, die Sie für Ihre Rezepte nicht brauchen. Die Beispiele in diesem Kapitel basieren auf dem Stack aus Erste Schritte mit Chef 11 Linux-Stacks, einem einfachen PHP-Anwendungsserver-Stack mit Load Balancer, PHP-Anwendungsservern und einem MySQL-Datenbankserver.

So erstellen Sie eine Stack-Konfiguration und ein Bereitstellungs-JSON

1. Erstellen Sie MyStack wie unter beschriebenErste Schritte mit Chef 11 Linux-Stacks, einschließlich der Bereitstellung von SimplePhpApp. Wenn Sie möchten, können Sie die zweite PHP App Server-Instanz weglassenSchritt 4: Skalieren MyStack, die in aufgerufen wird. Die Beispiele verwenden diese Attribute nicht.
2. Falls Sie das noch nicht getan haben, starten Sie die Instance `php-app1` und melden Sie sich über SSH an.
3. Führen Sie im Terminal-Fenster den folgenden agent cli-Befehl aus:

   ```
   sudo opsworks-agent-cli get_json
   ```

Über diesen Befehl werden die aktuellen Stack-Konfigurations- und Bereitstellungsattribute der Instance im JSON-Format im Terminal-Fenster aufgerufen.

4. Kopieren Sie die JSON in eine `.json`-Datei und speichern Sie diese lokal auf Ihrem Computer. Die Details sind abhängig vom verwendeten SSH-Client. Wenn Sie beispielsweise PuTTY unter Windows verwenden, können Sie mit dem Befehl `Copy All to Clipboard` den gesamten Text des Terminal-Fensters in die Windows-Zwischenablage kopieren. Dann können Sie den Inhalt in eine `.json`-Datei einfügen und nicht benötigten Text löschen.
5. Bearbeiten Sie MyStack JSON nach Bedarf. Es gibt zahlreiche Stack-Konfigurations- und Bereitstellungsattribute, von denen Rezeptbücher meist nur einen geringen Teil nutzen. Damit Ihre Umgebungsdatei übersichtlich bleibt, können Sie alle bis auf die von Ihren Rezeptbüchern tatsächlich verwendeten Attribute löschen, ohne die Struktur zu beschädigen.

   In diesem Beispiel wird eine stark bearbeitete Version von MyStack JSON verwendet, die nur zwei `['opsworks']['stack']` Attribute enthält, `['id]` und`['name']`. Erstellen Sie eine bearbeitete Version des MyStack JSON, die in etwa wie folgt aussieht:

```
{
  "opsworks": {
    "stack": {
      "name": "MyStack",
      "id": "42dfd151-6766-4f1c-9940-ba79e5220b58",
    },
  },
}
```

Um diese JSON in das Knotenobjekt der Instance einzufügen, müssen Sie es einer Test Kitchen-Umgebung hinzufügen.

So fügen Sie Stack-Konfigurations- und Bereitstellungsattribute der Test Kitchen-Umgebung hinzu

1. Erstellen Sie eine Umgebungsdatei `test.json` mit dem folgenden Inhalt und speichern Sie sie im Verzeichnis `environments` des Rezeptbuchs.

```
{
  "default_attributes": {
    "opsworks" : {
```

```
      "stack" : {
        "name" : "MyStack",
        "id" : "42dfd151-6766-4f1c-9940-ba79e5220b58"
      }
    }
  },
  "chef_type" : "environment",
  "json_class" : "Chef::Environment"
}
```

Die Umgebungsdatei besteht aus folgenden Elementen:

- `default_attributes`— Die Standardattribute im JSON-Format.

  Diese Attribute werden dem Knotenobjekt mit dem Attributtyp `default` hinzugefügt. Dieser Attributtyp wird von allen Stack-Konfigurations- und Bereitstellungs-JSON-Attributen verwendet. In diesem Beispiel wird die bereits vorgestellte bearbeitete Version der Stack-Konfigurations- und Bereitstellungs-JSON-Attribute verwendet.
- `chef_type`— Setze dieses Element auf`environment`.
- `json_class`— Setze dieses Element auf`Chef::Environment`.

2. Bearbeiten Sie `.kitchen.yml`, um wie nachfolgend beschrieben die Test Kitchen-Umgebung festzulegen.

```
---
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:
```

Sie können die Umgebung definieren, indem Sie der Standarddatei `.kitchen.yml`, die von `kitchen init` erstellt wurde, folgende Elemente hinzufügen.

provisioner

Fügen Sie die folgenden Elemente hinzu.

- `name`— Setze dieses Element auf`chef_solo`.

  Um die AWS OpsWorks Stacks-Umgebung besser zu replizieren, könnten Sie den lokalen Modus des Chef-Clients anstelle von Chef Solo verwenden. Der lokale Modus ist eine Chef-Client-Option, die auf einer abgespeckten Version von Chef Server (Chef Zero) basiert und lokal auf den Instances statt auf einem Remote-Server ausgeführt wird. So können Ihre Rezepte Chef Server-Funktionen wie die Suchfunktion oder Data Bags verwenden, ohne eine Verbindung zu einem Remote-Server herstellen zu müssen.
- `environments_path`— Das Cookbook-Unterverzeichnis, das die Umgebungsdatei enthält, `./environments` für dieses Beispiel.

suites:provisioner

Fügen Sie ein Element `solo_rb` ein, bei dem das Element `environment` den Namen der Umgebungsdatei (ohne die Erweiterung ".json") trägt. In diesem Beispiel wird für `environment` `test` verwendet.

3. Erstellen Sie eine Rezeptdatei `default.rb` mit folgendem Inhalt und speichern Sie sie im Verzeichnis `recipes` des Rezeptbuchs.

```
log "Stack name: #{node['opsworks']['stack']['name']}"
log "Stack id: #{node['opsworks']['stack']['id']}"
```

Dieses Rezept ruft nur die beiden Stack-Konfigurations- und Bereitstellungswerte ab, die Sie der Umgebung hinzugefügt haben. Obwohl das Rezept lokal in Virtual Box ausgeführt wird, referenzieren Sie diese Attribute mit derselben Knotensyntax, die Sie verwenden würden, wenn das Rezept auf einer AWS OpsWorks Stacks-Instanz ausgeführt würde.

4. Führen Sie `kitchen converge`. Es sollte etwa folgendes Protokoll ausgegeben werden.

```
...
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

...
```

Verwenden der Stack-Konfigurations- und Bereitstellungsattributwerte

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Rezepte benötigen häufig Informationen zur Stack-Konfiguration oder den bereitgestellten Apps. Sie könnten beispielsweise eine Liste der IP-Adressen des Stacks brauchen, um eine Konfigurationsdatei zu erstellen, oder das Bereitstellungsverzeichnis einer App zum Erstellen eines Protokollverzeichnisses. Anstatt diese Daten auf einem zentralen Server zu speichern, installiert AWS OpsWorks Stacks für jedes Lebenszyklusereignis eine Reihe von Stackkonfigurations- und Bereitstellungsattributen im Knotenobjekt jeder Instanz. Diese Attribute stellen den aktuellen Status des Stacks einschließlich der bereitgestellten Apps dar. Rezepte können benötige Daten aus dem Knotenobjekt abrufen.

> **Note**
>
> Anwendungen brauchen gelegentlich Informationen aus dem Knotenobjekt wie Stack-Konfigurations- und Bereitstellungsattributwerte. Anwendungen können jedoch nicht auf das Knotenobjekt zugreifen. Um einer Anwendung Daten aus einem Knotenobjekt bereitzustellen, können Sie ein Rezept implementieren, dass die benötigten Informationen aus dem Knotenobjekt abruft und in einer Datei in einem geeigneten Format speichert. Die Anwendung kann dann auf diese Datei zugreifen. Weitere Informationen sowie ein Beispiel finden Sie unter Übermitteln von Daten an Anwendungen.

Rezepte können wie nachfolgend beschrieben Stack-Konfigurations- und Bereitstellungsattributwerte aus Knotenobjekten abrufen.

- Direkt über den vollständig qualifizierten Namen eines Attributs

  Diese Methode kann auf beliebigen Linux-Stacks, nicht jedoch auf Windows-Stacks angewendet werden.
- Mit der Chef-Suche, über die Sie eine Anfrage für Attributwerte an das Knotenobjekt senden können

  Diese Methode ist für Windows-Stacks und Chef 11.10-Linux-Stacks geeignet.

> **Note**
>
> Für Linux-Stacks können Sie die Agenten-CLI verwenden, um eine Kopie der Stack-Konfigurations- und Bereitstellungsattribute einer Instance im JSON-Format zu erstellen. Weitere Informationen finden Sie unter Nachahmen der Stack-Konfiguration und Bereitstellungsattribute auf Vagrant.

Themen

Direktes Abrufen von Attributwerten

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Methode funktioniert nur auf Linux-Stacks.

Nachahmen der Stack-Konfiguration und Bereitstellungsattribute auf Vagrant zeigt, wie Sie Stack-Konfigurations- und Bereitstellungsdaten abrufen, indem Sie über die Knotensyntax direkt auf bestimmte Attribute verweisen. Dies ist manchmal die geeignetste Methode. Viele Attribute sind jedoch in Sammlungen oder Listen definiert, deren Inhalt und Name sich je nach Stack und im Laufe der Zeit auch einem bestimmten Stack unterscheiden können. Das Attribut `deploy` beispielsweise enthält eine Liste von App-Attributen, die nach dem kurzen Namen der App benannt sind. Die Liste einschließlich der App-Attributnamen unterscheidet sich meist je nach Stack und sogar je nach Bereitstellung.

Es kann oft hilfreich, wenn nicht sogar notwendig sein, die Attribute in einer Liste oder Sammlung durchzunummerieren, um die erforderlichen Daten abzurufen. Angenommen, Sie brauchen die öffentlichen IP-Adressen der Instances eines Stacks. Diese Information ist im Attribut `['opsworks']['layers']` gespeichert, bei dem es sich um eine Hash-Tabelle mit einem Element für jeden Layer des Stacks handelt, wobei die einzelnen Elemente nach den kurzen Namen der Layers benannt sind. Jedes Layer-Element besteht aus einer Hash-Tabelle, die die Attribute des Layers enthält, eines davon `['instances']`. Dieses Element wiederum enthält eine weitere Hash-Tabelle mit einem Attribut für jede Instance des Layers. Die Attribute sind hierbei nach den kurzen Namen der jeweiligen Instance benannt. Jedes Instance-Attribut enthält wiederum eine Hash-Tabelle mit den Attributen der Instance, darunter auch `['ip']` mit der öffentlichen IP-Adresse. Wenn Sie sich dies nur schwer vorstellen können, betrachten Sie das Beispiel im JSON-Format im folgenden Verfahren.

In diesem Beispiel wird gezeigt, wie Sie Daten aus der Stack-Konfigurations- und Bereitstellungs-JSON für die Layers eines Stacks abrufen.

So richten Sie das Rezeptbuch ein

1. Erstellen Sie ein Verzeichnis in `opsworks_cookbooks` namens `listip` und öffnen Sie es.
2. Initialisieren und konfigurieren Sie Test Kitchen wie unter Beispiel 1: Installieren von Paketen beschrieben.
3. Fügen Sie zwei Verzeichnisse zu `listip` hinzu: `recipes` und `environments`.
4. Erstellen Sie eine bearbeitete JSON-Version der MyStack Konfiguration und der Bereitstellungsattribute, die die relevanten Attribute enthält. Sie sollte etwa wie folgt aussehen.

```
{
  "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. Erstellen Sie eine Umgebungsdatei `test.json`, fügen Sie die Beispiel-JSON in `default_attributes` ein und speichern Sie die Datei im Verzeichnis `environments` des Rezeptbuchs. Die Datei sollte etwa wie folgt aussehen (der Kürze halber ist ein Großteil der Beispiel-JSON durch Ellipsen verkürzt dargestellt).

```
{
  "default_attributes" : {
    "opsworks": {
      "layers": {
        ...
      }
    }
  },
  "chef_type" : "environment",
  "json_class" : "Chef::Environment"
}
```

6. Ersetzen Sie den Text in `.kitchen.yml` durch folgenden.

```
---
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:
```

Nach dem Einrichten des Rezeptbuchs können Sie das folgende Rezept verwenden, um die Layer-IDs zu erfassen.

```
node['opsworks']['layers'].each do |layer, layerdata|
  log "#{layerdata['name']} : #{layerdata['id']}"
end
```

Das Rezept nummeriert die Layers in `['opsworks']['layers']` durch und speichert den Namen und die ID der einzelnen Layers.

So führen Sie das Rezept zum Erfassen der Layer-ID aus

1. Erstellen Sie eine Datei `default.rb` mit dem Beispielrezept und speichern Sie sie im Verzeichnis `recipes`.
2. Führen Sie `kitchen converge`.

Der relevante Teil der Ausgabe sollte etwa wie folgt aussehen.

```
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
```

Um die IP-Adressen der Instances aufzulisten, benötigen Sie eine verschachtelte Schleife wie nachfolgend beschrieben.

```
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
```

Die innere Schleife durchläuft die Instances jedes Layers und speichert die IP-Adressen.

So führen Sie das Rezept zum Speichern der Instance-IP-Adressen aus

1. Ersetzen Sie den Code in `default.rb` durch den Code aus dem Beispielrezept.
2. Führen Sie `kitchen converge` aus, um das Rezept auszuführen.

Der relevante Teil der Ausgabe sollte etwa wie folgt aussehen.

```
  * 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
```

Führen Sie anschließend `kitchen destroy` aus, da im nächsten Thema ein neues Rezeptbuch verwendet wird.

> **Note**
>
> Sammlungen von Stack-Konfigurations- und Bereitstellungs-JSON werden meist durchnummeriert, um Daten für eine bestimmte bereitgestellte App wie beispielsweise das Bereitstellungsverzeichnis abzurufen. Ein Beispiel finden Sie unter Bereitstellungsrezepte.

Abrufen von Attributwerten mit der Chef-Suche

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Methode ist für Windows-Stacks und Chef 11.10-Linux-Stacks verfügbar.

Es kann kompliziert sein, Stack-Konfigurations- und Bereitstellungsattributwerte direkt aus dem Knotenobjekt abzurufen. Für Windows-Stacks ist dies generell nicht möglich. Alternativ können Sie mit der Chef-Suche die benötigten Attribute abrufen. Wenn Sie mit dem Chef-Server vertraut sind, werden Sie feststellen, dass die Chef-Suche mit AWS OpsWorks Stacks etwas anders funktioniert. Da AWS OpsWorks Stacks Chef-Client im lokalen Modus verwendet, hängt die Chef-Suche von einer lokalen Version des Chef-Servers namens chef-zero ab, sodass die Suche auf den Daten basiert, die lokal im Knotenobjekt der Instanz gespeichert sind, und nicht auf einem Remote-Server.

In der Praxis spielt es normalerweise keine Rolle, die Suche auf lokal gespeicherte Daten zu beschränken, da das Knotenobjekt auf einer AWS OpsWorks Stacks-Instanz die Stack-Konfiguration und die Bereitstellungsattribute enthält. Sie enthalten die meisten, wenn nicht sogar alle Daten, die Rezepte normalerweise vom Chef-Server beziehen würden, und verwenden dieselben Namen, sodass Sie in der Regel den für den Chef-Server geschriebenen Suchcode auf AWS OpsWorks Stacks-Instanzen ohne Änderung verwenden können. Weitere Informationen finden Sie unter Verwenden der Chef-Suchfunktion.

Nachfolgend finden Sie die Basisstruktur einer Suchanfrage:

```
result = search(:search_index, "key:pattern")
```

- Der Suchindex gibt an, welche Attribute abgefragt werden, und legt fest, welcher Objekttyp zurückgegeben wird.
- Der Schlüssel gibt den Attributnamen an.
- Über das Muster wird festgelegt, welche Werte des Attributs abgerufen werden sollen.

  Sie können bestimmte Attributwerte abrufen oder mithilfe von Platzhaltern einen Wertebereich abfragen.
- Als Ergebnis erhalten Sie eine Liste mit Objekten, die der Suchanfrage entsprechen. Jedes Ergebnis ist hierbei eine Hash-Tabelle mit mehreren zusammengehörigen Attributen.

  Wenn Sie beispielsweise den Suchindex `node` verwenden, gibt die Suchanfrage eine Liste der Instance-Objekte für alle Instances zurück, die der Suchanfrage entsprechen. Jedes Objekt in einer

Hash-Tabelle enthält eine Reihe von Attributen, die die Konfiguration einer Instance festlegen, beispielsweise Hostname und IP-Adresse.

In der folgenden Anfrage wird beispielsweise der Suchindex `node` verwendet. Hierbei handelt es sich um einen Standard-Chef-Index, der auf die Stack-Instances (in Chef-Terminologie "Knoten") angewendet wird. Er sucht nach Instances mit dem Hostname `myhost`.

```
result = search(:node, "hostname:myhost")
```

Als Suchergebnis erhalten Sie eine Liste von Instance-Objekten mit dem Hostnamen `myhost`. Wenn Sie beispielsweise das Betriebssystem der ersten Instance benötigen, ist dieses unter `result[0][:os]` gespeichert. Wenn die Suchanfrage mehrere Objekte zurückgibt, können Sie diese durchnummerieren, um die benötigten Informationen zu erhalten.

Wie Sie die Suche in einem Rezept genau einsetzen, hängt davon ab, ob Sie einen Linux- oder Windows-Stack verwenden. In den folgenden Themen finden Sie Beispiele für beide Stack-Typen.

Themen

- Verwenden der Suche auf einem Linux-Stack
- Verwenden der Suche auf einem Windows-Stack

Verwenden der Suche auf einem Linux-Stack

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Dieses Beispiel basiert auf einem Linux-Stack mit einem einzelnen PHP-Anwendungsserver. Die öffentliche IP-Adresse des Servers wird mithilfe der Chef-Suche abgerufen und dann in einer Datei im Verzeichnis `/tmp` gespeichert. Im Grunde werden dieselben Informationen aus dem Knotenobjekt abgerufen wie mit Direktes Abrufen von Attributwerten , der Code ist jedoch wesentlich simpler und unabhängig von der genauen Struktur der Stack-Konfigurations- und Bereitstellungsattribute.

Nachfolgend wird kurz beschrieben, wie Sie für dieses Beispiel den Stack erstellen. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

> **Note**
>
> Wenn du noch kein benutzerdefiniertes Rezept auf einer AWS OpsWorks Stacks-Instanz ausgeführt hast, solltest du zuerst das Beispiel durchgehen. Ausführen eines Rezepts auf einer Linux-Instance

Erstellen eines Stacks

1. Öffnen Sie die AWS OpsWorks Stacks-Konsole und klicken Sie auf Add Stack (Stack hinzufügen).
2. Legen Sie die folgenden Einstellungen fest, übernehmen Sie für die restlichen Einstellungen die Standardwerte und klicken Sie auf Add Stack (Stack hinzufügen).

   - Name — searchJSON
   - Standard-SSH-Schlüssel — Ein Amazon EC2 EC2-Schlüsselpaar

   Wenn Sie ein Amazon EC2 EC2-Schlüsselpaar erstellen müssen, finden Sie weitere Informationen unter Amazon EC2 EC2-Schlüsselpaare. Das Schlüsselpaar muss sich in derselben AWS-Region befinden wie die Instance. Das Beispiel verwendet die Region USA West (Oregon).
3. Klicken Sie auf Layer hinzufügen und fügen Sie dem Stack einen PHP App Server-Layer mit Standardeinstellungen hinzu.
4. Fügen Sie dem Layer eine 24/7-Instance mit den Standardeinstellungen hinzu und starten Sie sie.

So richten Sie das Rezeptbuch ein

1. Erstellen Sie ein Verzeichnis in `opsworks_cookbooks` namens `searchjson` und öffnen Sie es.
2. Erstellen Sie eine Datei `metadata.rb` mit dem folgenden Inhalt und speichern Sie sie unter `opstest`.

```
name "searchjson"
version "0.1.0"
```

3. Erstellen Sie ein Verzeichnis `recipes` in `searchjson`.
4. Erstellen Sie eine Datei `default.rb` mit dem folgenden Rezept und speichern Sie sie im Verzeichnis `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
```

Auf Linux-Stacks wird nur der Suchindex `node` unterstützt. Das Rezept ruft mithilfe dieses Index eine Liste der Instances im Layer `php-app` ab. Da der Layer ja nur eine Instance hat, weist das Rezept diese einfach `phpserver` zu. Wenn der Layer mehrere Instances hätte, könnten Sie diese durchnummerieren, um die benötigten Informationen abzurufen. Jedes Listenelement ist eine Hash-Tabelle mit einer Reihe von Instance-Attributen. Das Attribut `ip` enthält die öffentliche IP-Adresse der Instance. Sie können die Adresse also im nachfolgenden Rezeptcode als `phpserver[:ip]` darstellen.

Nachdem Sie eine Nachricht zum Chef-Protokoll hinzugefügt haben, verwendet das Rezept eine `file`-Ressource, um eine Datei mit dem Namen `ip_addresses` zu erstellen. Das Attribut `content` stellt `phpserver[:ip]` als Zeichenfolge dar. Wenn Chef die Datei `ip_addresses` erstellt, wird diese Zeichenfolge in der Datei gespeichert.

5. Erstellen Sie ein `.zip` Archiv von`opsworks_cookbooks`, laden Sie das Archiv in einen Amazon S3 S3-Bucket hoch, machen Sie das Archiv öffentlich und notieren Sie die URL des Archivs. Weitere Informationen zu Rezeptbuch-Repositorys finden Sie unter Rezeptbuch-Repositorys.

   Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Jetzt können Sie das Rezeptbuch installieren und das Rezept ausführen.

So führen Sie das Rezept aus

1. Bearbeiten Sie den Stack, um benutzerdefinierte Rezeptbücher zu aktivieren, und legen Sie folgende Einstellungen fest:

   - Repository-Typ — HTTP-Archiv
   - Repository-URL — Die URL des Kochbuch-Archivs, die Sie zuvor aufgezeichnet haben

   Verwenden Sie für die übrigen Einstellungen die Standardwerte und klicken Sie auf Save (Speichern), um die Stack-Konfiguration zu aktualisieren und zu speichern.
2. Bearbeiten Sie die benutzerdefinierte Layer-Konfiguration und weisen Sie `searchjson::default` sie dem Setup-Ereignis der Ebene zu. AWS OpsWorks Stacks führt das Rezept aus, nachdem die Instanz gestartet wurde oder wenn Sie das Setup-Ereignis explizit auslösen.
3. Führen Sie den Stack-Befehl „Update Custom Cookbooks” aus, um die aktuelle Version Ihres benutzerdefinierten Rezeptbuch-Repository auf den Stack-Instances zu installieren. Wenn bereits eine ältere Version des Repositorys installiert ist, wird diese überschrieben.
4. Führen Sie das Rezept aus, indem Sie den Stack-Befehl Setup ausführen. Dadurch wird ein Einrichtungsereignis auf der Instance ausgelöst und `searchjson::default` wird ausgeführt. Lassen Sie die Seite Running command setup page offen.

Nachdem das Rezept erfolgreich ausgeführt wurde, können Sie es überprüfen.

Sie überprüfen Sie searchjson

1. Sehen Sie sich zunächst das Chef-Protokoll und das letzte Einrichtungsereignis darin an. Klicken Sie auf der Seite Running command setup page auf show in der Spalte Log der Instance „php-app1”, um das Protokoll anzuzeigen. Blättern Sie nach unten zu Ihrer Protokollnachricht in der Mitte. Diese sieht etwa wie folgt aus.

```
...
[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. Melden Sie sich über SSH bei der Instance an und rufen Sie den Inhalt des Verzeichnisses `/tmp` auf. Dieses sollte eine Datei `ip_addresses` mit der IP-Adresse enthalten.

Verwenden der Suche auf einem Windows-Stack

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks bietet zwei Optionen, um die Suche in Windows-Stacks zu verwenden.

- Den Suchindex `node`, mit dem eine Reihe von Standard-Chef-Attributen abgerufen werden kann

  Wenn Sie bereits Rezepte mit verwendetem Suchcode habe`node`n, funktionieren diese normalerweise ohne Änderung auf AWS OpsWorks Stacks-Stacks.
- Weitere Suchindexe, mit denen eine Reihe AWS OpsWorks Stacks-spezifischer Attribute sowie einige Standardattribute abgerufen werden können

  Diese Indexe werden in Verwenden von AWS OpsWorks Stacks-spezifischen Suchindizes auf Windows Stacks näher erläutert.

Wir empfehlen, Standardinformationen wie Hostnamen und IP-Adressen mit `node` abzurufen. So sind Ihre Rezepte kompatibel mit der Standard-Chef-Vorgehensweise. Verwenden Sie die AWS OpsWorks Stacks-Suchindexe, um Informationen abzurufen, die für Stacks spezifisch sind. AWS OpsWorks

Themen

- Verwenden des Knoten-Suchindexes auf Windows-Stacks
- Verwenden von AWS OpsWorks Stacks-spezifischen Suchindizes auf Windows Stacks

Verwenden des Knoten-Suchindexes auf Windows-Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> In diesem Beispiel wird davon ausgegangen, dass Sie das Beispiel Ausführen eines Rezepts auf einer Windows-Instance bereits durchgearbeitet haben. Falls Sie das noch nicht getan haben, holen Sie das nun nach. Insbesondere wird darin beschrieben, wie Sie für Ihre Instances RDP-Zugriff aktivieren.

Dieses Beispiel basiert auf einem Windows-Stack mit einem benutzerdefinierten Layer und einer Instance. Es verwendet die Chef-Suche mit dem Suchindex `node`, um die öffentliche IP-Adresse des Servers abzurufen, und speichert die Adresse in einer Datei im Verzeichnis `C:\tmp`. Nachfolgend wird kurz beschrieben, wie Sie für dieses Beispiel den Stack erstellen. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

Erstellen eines Stacks

1. Öffnen Sie die AWS OpsWorks Stacks-Konsole und wählen Sie Add Stack (Stack hinzufügen) aus.
2. Legen Sie die folgenden Einstellungen fest, übernehmen Sie für die restlichen Einstellungen die Standardwerte und wählen Sie Add Stack (Stack hinzufügen) aus.
   - Name — NodeSearch
   - Region — USA West (Oregon)

Dieses Beispiel funktioniert in jeder Region, wir empfehlen jedoch, US West (Oregon) für Tutorials zu verwenden.

- Standardbetriebssystem — Microsoft Windows Server 2012 R2

3. Wählen Sie Add a layer (Layer hinzufügen) aus und fügen Sie dem Stack einen benutzerdefinierten Layer mit folgenden Einstellungen hinzu:

   - Name — IPTest
   - Kurzname — iptest

4. Fügen Sie eine 24/7-t2.Micro-Instance mit den Standardeinstellungen zum IPTest-Layer hinzu und starten Sie sie. Sie hat den Namen iptest1.

   AWS OpsWorks Stacks weist diese Instanz automatisch `AWS-OpsWorks-RDP-Server` zu, sodass sich autorisierte Benutzer bei der Instanz anmelden können.

5. Wählen Sie Permissions (Berechtigungen), dann Edit (Bearbeiten) und anschließend SSH/RDP und sudo/admin aus. Reguläre Benutzer benötigen zusätzlich zur Sicherheitsgruppe `AWS-OpsWorks-RDP-Server` diese Autorisierung, um sich bei der Instance anzumelden.

> **Note**
>
> Sie können sich auch als Administrator anmelden, allerdings mit einem anderen Verfahren. Weitere Informationen finden Sie unter Anmelden mit RDP.

So richten Sie das Rezeptbuch ein

1. Erstellen Sie ein Verzeichnis `nodesearch` und öffnen Sie es.
2. Erstellen Sie eine Datei `metadata.rb` mit dem folgenden Inhalt und speichern Sie sie unter `opstest`.

```
name "nodesearch"
version "0.1.0"
```

3. Erstellen Sie ein Verzeichnis `recipes` in `nodesearch`.
4. Erstellen Sie eine Datei `default.rb` mit dem folgenden Rezept und speichern Sie sie im Verzeichnis `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
```

Vom Rezept werden folgende Schritte ausgeführt:

1. Erstellen Sie mithilfe einer Verzeichnisressource für die Datei ein Verzeichnis `C:\tmp`.

   Weitere Informationen zu dieser Ressource finden Sie unter Beispiel 3: Erstellen von Verzeichnissen.

2. Verwendet die Chefsuche mit dem Suchindex `node`, um eine Liste der Knoten (Instances) mit einem Hostnamen abzurufen, der mit `iptest` beginnt.

   Wenn Sie das Standarddesign verwenden, bei dem Hostnamen durch Anhängen von Ganzzahlen an den kurzen Namen des Layers erstellt werden, gibt diese Suchanfrage alle Instances im IPTest-Layer zurück. In diesem Beispiel hat der Layer nur eine Instance, daher weist das Rezept diese einfach `windowsserver` zu. Wenn mehrere Instances vorhanden sind, können Sie die vollständige Liste abrufen und durchnummerieren.

3. Fügt dem Chef-Protokoll für diesen Durchlauf eine Nachricht mit der IP-Adresse hinzu.

   Das Objekt `windowsserver` ist eine Hash-Tabelle, bei dem das Attribut `ipaddress` die öffentliche IP-Adresse der Instance enthält. Sie können diese Adresse im folgenden Rezept daher als `windowsserver[:ipaddress]` bezeichnen. Das Rezept fügt die entsprechende Zeichenfolge in die Nachricht ein und fügt diese dem Chef-Protokoll hinzu.

4. Verwendet die Ressource `file`, um eine Datei `C:\tmp\addresses.txt` mit der IP-Adresse zu erstellen.

Über das Attribut `content` wird der Inhalt der Datei, in diesem Fall die öffentliche IP-Adresse, festgelegt.

5. Erstellen Sie ein `.zip`-Archiv von `nodesearch`, laden Sie das Archiv in einen S3-Bucket hoch, veröffentlichen Sie das Archiv und notieren Sie sich die URL des Archivs.

   Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Jetzt können Sie das Rezeptbuch installieren und das Rezept ausführen.

So installieren Sie das Rezeptbuch und führen das Rezept aus

1. Bearbeiten Sie den Stack, um benutzerdefinierte Rezeptbücher zu aktivieren, und legen Sie folgende Einstellungen fest:

   - Repository-Typ — S3-Archiv
   - Repository-URL — Die URL des Kochbuch-Archivs, die Sie zuvor aufgezeichnet haben

   Übernehmen Sie für die übrigen Einstellungen die Standardwerte und wählen Sie Save aus, um die Stack-Konfiguration zu aktualisieren und zu speichern.
2. Führen Sie den Stack-Befehl „Update Custom Cookbooks” aus, um die aktuelle Version Ihrer benutzerdefinierten Rezeptbücher auf den Stack-Instances einschließlich Online-Instances zu installieren. Wenn bereits eine ältere Version der Rezeptbücher installiert ist, werden diese überschrieben.
3. Nachdem die benutzerdefinierten Rezeptbücher aktualisiert wurden, führen Sie das Rezept mithilfe des Stack-Befehls Execute Recipes aus. Achten Sie darauf, dass bei Recipes to execute **`nodesearch::default`** eingestellt ist. Durch diesen Befehl wird Chef mit Ihrem Rezept ausgeführt. Lassen Sie die Seite "execute_recipes" offen.

Nachdem das Rezept erfolgreich ausgeführt wurde, können Sie es überprüfen.

So überprüfen Sie nodesearch

1. Rufen Sie das Chef-Protokoll auf und sehen Sie sich das letzte execute_recipes-Ereignis an. Wählen Sie auf der Seite Running command execute_recipes page die Option show in der

Spalte Log der Instance „iptest1" aus, um das Protokoll anzuzeigen. Blättern Sie nach unten zu Ihrer Protokollnachricht am Ende. Diese sieht etwa wie folgt aus.

```
...
[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. Melden Sie sich mit RDP bei der Instance an und rufen Sie das Verzeichnis `C:\tmp\addresses.txt` auf.

Verwenden von AWS OpsWorks Stacks-spezifischen Suchindizes auf Windows Stacks

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

In diesem Beispiel wird davon ausgegangen, dass Sie das Beispiel Ausführen eines Rezepts auf einer Windows-Instance bereits durchgearbeitet haben. Falls Sie das noch nicht getan haben, holen Sie das nun nach. Insbesondere wird darin beschrieben, wie Sie für Ihre Instances RDP-Zugriff aktivieren.

AWS OpsWorks Stacks bietet zusätzlich zu den folgenden Suchindizes: node

- `aws_opsworks_stack`— Die Stack-Konfiguration.
- `aws_opsworks_layer`— Die Layer-Konfigurationen des Stacks.
- `aws_opsworks_instance`— Die Instanzkonfigurationen des Stacks.
- `aws_opsworks_app`— Die App-Konfigurationen des Stacks.
- `aws_opsworks_user`— Die Benutzerkonfigurationen des Stacks.
- `aws_opsworks_rds_db_instance`— Verbindungsinformationen für registrierte RDS-Instances.

Diese Indizes enthalten einige Standard-Chef-Attribute, sind jedoch hauptsächlich für das Abrufen von AWS OpsWorks Stacks-spezifischen Attributen vorgesehen. Beispielsweise enthält `aws_opsworks_instance` ein Attribut `status`, das den Status der Instance angibt, z. B. `online`.

> **Note**
>
> Es wird empfohlen, nach Möglichkeit node zu verwenden, um Ihre Rezepte mit den Chef-Standards kompatibel zu halten. Ein Beispiel finden Sie unter Verwenden des Knoten-Suchindexes auf Windows-Stacks.

Dieses Beispiel zeigt, wie die AWS OpsWorks Stacks-Indizes verwendet werden, um den Wert eines Stacks-spezifischen Attributs abzurufen. AWS OpsWorks Das Beispiel basiert auf einem einfachen Windows-Stack mit einem benutzerdefinierten Layer und einer Instances. Es verwendet die Chef-Suche, um die AWS OpsWorks Stacks-ID der Instanz abzurufen, und fügt die Ergebnisse in das Chef-Protokoll ein.

Nachfolgend wird kurz beschrieben, wie Sie für dieses Beispiel einen Stack erstellen. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

Erstellen eines Stacks

1. Öffnen Sie die AWS OpsWorks Stacks-Konsole und wählen Sie + Stack aus. Legen Sie die folgenden Einstellungen fest, übernehmen Sie für die restlichen Einstellungen die Standardwerte und wählen Sie Add Stack (Stack hinzufügen) aus.

   - Name — IDSearch
   - Region — USA West (Oregon)

Dieses Beispiel funktioniert in jeder Region, wir empfehlen jedoch, US West (Oregon) für Tutorials zu verwenden.

- Standardbetriebssystem — Microsoft Windows Server 2012 R2

2. Wählen Sie Add a layer (Layer hinzufügen) aus und fügen Sie dem Stack einen benutzerdefinierten Layer mit folgenden Einstellungen hinzu:

   - Name — IDCheck
   - Kurzname — idcheck

3. Fügen Sie eine 24/7-t2.Micro-Instance mit den Standardeinstellungen zum IDCheck-Layer hinzu und starten Sie sie. Sie hat den Namen iptest1.

   AWS OpsWorks Stacks wird dieser Instanz automatisch zugewiesen`AWS-OpsWorks-RDP-Server`. Aktivieren von RDP-Zugrifferklärt, wie dieser Sicherheitsgruppe eine Regel für eingehenden Datenverkehr hinzugefügt wird, die es autorisierten Benutzern ermöglicht, sich bei der Instanz anzumelden.

4. Wählen Sie Permissions (Berechtigungen), dann Edit (Bearbeiten) und anschließend SSH/RDP und sudo/admin aus. Reguläre Benutzer benötigen zusätzlich zur Sicherheitsgruppe `AWS-OpsWorks-RDP-Server` diese Autorisierung, um sich bei der Instance anzumelden.

> **Note**
>
> Sie können sich auch als Administrator anmelden, allerdings mit einem anderen Verfahren. Weitere Informationen finden Sie unter Anmelden mit RDP.

So richten Sie das Rezeptbuch ein

1. Erstellen Sie ein Verzeichnis `idcheck` und öffnen Sie es.
2. Erstellen Sie eine Datei `metadata.rb` mit dem folgenden Inhalt und speichern Sie sie unter `opstest`.

```
name "idcheck"
version "0.1.0"
```

3. Erstellen Sie ein Unterverzeichnis `recipes` unter `idcheck` und fügen Sie dem Verzeichnis eine Datei `default.rb` mit folgendem Rezept hinzu.

```
windowsserver = search(:aws_opsworks_instance, "hostname:idcheck*").first
Chef::Log.info("**********The public IP address is:
 '#{windowsserver[:instance_id]}'**********")
```

Das Rezept verwendet die Chef-Suche mit einem `aws_opsworks_instance`-Suchindex, um die Instance-Attribute aller Instances im Stack mit einem Hostnamen abzurufen, der mit `idcheck` beginnt. Wenn Sie das Standarddesign verwenden, bei dem Hostnamen durch Anhängen von Ganzzahlen an den kurzen Namen des Layers erstellt werden, gibt diese Suchanfrage alle Instances im IDCheck-Layer zurück. In diesem Beispiel hat der Layer nur eine Instance, daher weist das Rezept diese einfach `windowsserver` zu. Wenn mehrere Instances vorhanden sind, können Sie die vollständige Liste abrufen und durchnummerieren.

Das Rezept macht sich die Tatsache zunutze, dass der Stack nur eine Instance mit diesem Hostnamen enthält und bereits das erste Ergebnis das richtige ist. Wenn ein Stack mehrere Instances enthält, erhalten Sie bei der Suche nach anderen Attributen möglicherweise mehrere Ergebnisse. Eine Liste der Instance-Attribute finden Sie unter Data Bag für Instances (aws_opsworks_instance).

Die Instanzattribute sind im Grunde eine Hashtabelle, und die AWS OpsWorks Stacks-ID der Instanz ist dem `instance_id` Attribut zugewiesen, sodass Sie die ID als bezeichnen können. `windowsserver[:instance_id]` Das Rezept fügt die entsprechende Zeichenfolge in die Nachricht ein und fügt diese dem Chef-Protokoll hinzu.

4. Erstellen Sie ein `.zip` Archiv des `ipaddress` Kochbuches, laden Sie das Archiv in einen Amazon S3 S3-Bucket hoch und notieren Sie sich die URL des Archivs. Weitere Informationen zu Rezeptbuch-Repositorys finden Sie unter Rezeptbuch-Repositorys.

   Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Jetzt können Sie das Rezeptbuch installieren und das Rezept ausführen.

So installieren Sie das Rezeptbuch und führen das Rezept aus

1. Bearbeiten Sie den Stack, um benutzerdefinierte Rezeptbücher zu aktivieren, und legen Sie folgende Einstellungen fest:

- Repository-Typ — S3-Archiv
- Repository-URL — Die URL des Kochbuch-Archivs, die Sie zuvor aufgezeichnet haben

Übernehmen Sie für die übrigen Einstellungen die Standardwerte und wählen Sie Save aus, um die Stack-Konfiguration zu aktualisieren und zu speichern.

2. Führen Sie den Stack-Befehl „Update Custom Cookbooks" aus, um die aktuelle Version Ihrer benutzerdefinierten Rezeptbücher auf den Stack-Instances einschließlich Online-Instances zu installieren. Wenn bereits eine ältere Version der Rezeptbücher installiert ist, werden diese überschrieben.
3. Nachdem die benutzerdefinierten Rezeptbücher aktualisiert wurden, führen Sie das Rezept mithilfe des Stack-Befehls Execute Recipes aus. Achten Sie darauf, dass bei Recipes to execute **`idcheck::default`** eingestellt ist. Durch diesen Befehl wird Chef mit Ihrem Rezept ausgeführt. Lassen Sie die Seite "execute_recipes" offen.

Nachdem das Rezept erfolgreich ausgeführt wurde, können Sie dies im Chef-Protokoll für das letzte execute_recipes-Ereignis überprüfen. Wählen Sie auf der Seite Running command execute_recipes page die Option show in der Spalte Log der Instance „iptest1" aus, um das Protokoll anzuzeigen. Blättern Sie nach unten zu Ihrer Protokollnachricht am Ende. Diese sieht etwa wie folgt aus.

```
...
[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
...
```

Verwenden von externen Rezeptbüchern auf einer Linux-Instance: Berkshelf

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Berkshelf ist nur für Chef 11.10-Linux-Stacks verfügbar.

Bevor Sie mit der Implementierung eines Rezeptbuchs beginnen, sollten Sie sich die Seite Chef Community Cookbooks ansehen. Hier finden Sie Rezeptbücher, die von Mitgliedern der Chef-Community für die unterschiedlichsten Zwecke erstellt wurden. Viele dieser Kochbücher können ohne Änderungen mit AWS OpsWorks Stacks verwendet werden, sodass Sie sie möglicherweise für einige Ihrer Aufgaben nutzen können, anstatt den gesamten Code selbst zu implementieren.

Um externe Rezeptbücher auf Instances verwenden zu können, müssen Sie es zunächst installieren und seine Abhängigkeiten verwalten können. Am einfachsten verwenden Sie dafür den Abhängigkeitsmanager Berkshelf. Berkshelf funktioniert auf Amazon EC2 EC2-Instances, einschließlich AWS OpsWorks Stacks-Instances, ist aber auch für die Zusammenarbeit mit Test Kitchen und Vagrant konzipiert. Die Verwendung auf Vagrant ist jedoch etwas anders als bei AWS OpsWorks Stacks, sodass dieses Thema Beispiele für beide Plattformen enthält. Weitere Informationen zur Verwendung von Berkshelf finden Sie unter Berkshelf.

Themen

- Verwenden von Berkshelf mit Test Kitchen und Vagrant
- Berkshelf mit Stacks verwenden AWS OpsWorks

Verwenden von Berkshelf mit Test Kitchen und Vagrant

In diesem Beispiel wird erläutert, wie Sie mit Berkshelf das Community-Rezeptbuch "getting-started" installieren und das darin enthaltene Rezept ausführen, um eine kurze Textdatei im Home-Verzeichnis der Instance zu installieren.

So installieren Sie Berkshelf und initialisieren ein Rezeptbuch

1. Installieren Sie das Berkshelf-Gem wie folgt auf Ihrem Computer.

```
gem install berkshelf
```

Abhängig von Ihrer Workstation benötigen Sie hierfür womöglich `sudo` oder einen Ruby-Umgebungsmanager wie RVM. Führen Sie `berks --version` aus, um zu überprüfen, ob Berkshelf korrekt installiert wurde.

2. Das Rezeptbuch für dieses Thema heißt "external_cookbook". Sie können mit Berkshelf ein initialisiertes Rezeptbuch erstellen, statt wie in den vorherigen Themen manuell vorzugehen. Wechseln Sie hierfür ins Verzeichnis `opsworks_cookbooks` und führen Sie den folgenden Befehl aus.

   ```
   berks cookbook external_cookbook
   ```

   Der Befehl erstellt das Verzeichnis `external_cookbook` sowie einige Standardunterverzeichnisse von Chef und Test Kitchen, darunter `recipes` und `test`. Außerdem erstellt er Standardversionen einiger Standarddateien, darunter folgende:

   - `metadata.rb`
   - Konfigurationsdateien für Vagrant, Test Kitchen und Berkshelf
   - Ein leeres Rezept `default.rb` im Verzeichnis `recipes`

   > **Note**
   >
   > Sie müssen `kitchen init` nicht ausführen, da der Befehl `berks cookbook` diese Aufgaben bereits ausführt.

3. Führen Sie `kitchen converge`. Das neu erstellte Rezeptbuch hat bisher noch keine relevante Funktion, kommt der Sache aber schon nahe.

> **Note**
>
> Sie können mithilfe von `berks init` auch ein vorhandenes Rezeptbuch initialisieren, um Berkshelf zu verwenden.

Um mithilfe von Berkshelf die externen Abhängigkeiten eines Rezeptbuchs zu verwalten, muss das Stammverzeichnis des Rezeptbuchs eine Datei `Berksfile` enthalten. Dies ist eine Konfigurationsdatei, in der festgelegt ist, wie Berkshelf Abhängigkeiten verwaltet. Wenn Sie

mithilfe von `berks cookbook` das Rezeptbuch `external_cookbook` erstellen, wird eine Datei `Berksfile` mit folgendem Inhalt angelegt.

```
source "https://supermarket.chef.io"
metadata
```

Diese Datei hat folgende Deklarationen:

- `source`— Die URL einer Kochbuchquelle.

  Eine Berksfile-Datei kann beliebig viele `source`-Deklarationen enthalten, von denen jede eine Standardquelle für abhängige Rezeptbücher angibt. Wenn Sie nicht explizit eine Rezeptbuchquelle angeben, durchsucht Berkshelf die Standard-Repositorys nach einem Rezeptbuch mit demselben Namen. Die Standard-Berksfile-Datei enthält ein einzelnes Attribut `source`, das auf das Community-Rezeptbuch-Repository verweist. Dieses Repository enthält das Rezeptbuch "getting-started", Sie können diese Zeile also unverändert lassen.
- `metadata`— Weist Berkshelf an, Kochbuch-Abhängigkeiten aufzunehmen, die in der Kochbuchdatei deklariert sind. `metadata.rb`

  Mithilfe des Attributs `cookbook` können Sie auch selbst ein abhängiges Rezeptbuch in der Berksfile-Datei angeben. Weitere Informationen dazu erhalten Sie im weiteren Verlauf dieses Themas.

Es gibt zwei Möglichkeiten, die Abhängigkeiten eines Rezeptbuchs zu deklarieren:

- Fügen Sie eine `cookbook`-Deklaration in die Berksfile-Datei ein.

  Dies ist der Ansatz, der von Stacks verwendet wird. AWS OpsWorks Fügen Sie beispielsweise `cookbook "getting-started"` in die Berksfile-Datei ein, um das für dieses Beispiel benötigte Rezeptbuch "getting-started" festzulegen. Berkshelf durchsucht daraufhin die Standard-Repositorys nach einem Rezeptbuch mit diesem Namen. Sie können mithilfe von `cookbook` auch eine genaue Rezeptbuchquelle und sogar eine bestimmte Version festlegen. Weitere Informationen finden Sie unter Berkshelf.
- Fügen Sie eine `metadata`-Deklaration in die Berksfile-Datei ein und deklarieren Sie die Abhängigkeit in `metadata.rb`.

Über diese Deklaration weisen Sie Berkshelf an, die Abhängigkeiten des Rezeptbuchs aus der Datei `metadata.rb` ebenfalls zu installieren. Um beispielsweise eine der Abhängigkeiten des Rezeptbuchs "getting-started" zu deklarieren, fügen Sie eine `depends 'getting-started'`-Deklaration in die Datei `metadata.rb` des Rezeptbuchs ein.

In diesem Beispiel wird aus Gründen der Konsistenz mit AWS OpsWorks Stacks der erste Ansatz verwendet.

So installieren Sie das Rezeptbuch "getting-started"

1. Bearbeiten Sie die Standard-Berksfile-Datei und ersetzen Sie die `metadata`-Deklaration durch eine `cookbook`-Deklaration für `getting-started`. Der Inhalt sollte wie folgt aussehen.

   ```
   source "https://supermarket.chef.io"

   cookbook 'getting-started'
   ```

2. Führen Sie `berks install` aus, um das Rezeptbuch "getting-started" aus dem Community-Rezeptbuch-Repository in Ihrem lokalen Berkshelf-Verzeichnis, normalerweise `~/.berkshelf`, zu installieren. Dieses Verzeichnis wird oftmals einfach als das Berkshelf bezeichnet. Im Verzeichnis `cookbooks` von Berkshelf sollten Sie nun das Verzeichnis für das Rezeptbuch "" mit dem Namen `getting-started-0.4.0`getting-started- ((oder ähnlich) finden.
3. Ersetzen Sie `external_cookbook::default` in der Ausführungsliste `.kitchen.yml` durch `getting-started::default`. In diesem Beispiel werden keine Rezepte aus "external_cookbook" ausgeführt. Es wird nur benötigt, um das Rezeptbuch "getting-started" zu verwenden. Die Datei `.kitchen.yml` sollte jetzt wie folgt aussehen.

   ```
   ---
   driver:
     name: vagrant

   provisioner:
     name: chef_solo

   platforms:
     - name: ubuntu-12.04
   ```

```
suites:
  - name: default
    run_list:
      - recipe[getting-started::default]
    attributes:
```

4. Führen Sie `kitchen converge` aus und melden Sie sich mit `kitchen login` bei der Instance an. Das Anmeldeverzeichnis sollte eine Datei `chef-getting-started.txt` mit etwa folgendem Inhalt enthalten:

```
Welcome to Chef!

This is Chef version 11.12.8.
Running on ubuntu.
Version 12.04.
```

   Test Kitchen installiert Rezeptbücher im Verzeichnis `/tmp/kitchen/cookbooks` der Instance. Wenn Sie den Inhalt dieses Verzeichnisses aufrufen, sehen Sie zwei Rezeptbücher: "external_cookbook" und "getting-started".

5. Führen Sie `kitchen destroy` aus, um die Instance herunterzufahren. Das nächste Beispiel verwendet eine AWS OpsWorks Stacks-Instanz.

Berkshelf mit Stacks verwenden AWS OpsWorks

AWS OpsWorks Stacks unterstützt optional Stacks von Berkshelf für Chef 11.10. Um Berkshelf für Ihren Stack zu verwenden, gehen Sie wie folgt vor.

- Aktivieren Sie Berkshelf für den Stack.

  AWS OpsWorks Stacks kümmert sich dann um die Details der Installation von Berkshelf auf den Instanzen des Stacks.
- Fügen Sie dem Stammverzeichnis Ihres Rezeptbuch-Repositorys eine Berksfile-Datei hinzu.

  Die Berksfile-Datei muss für alle abhängigen Rezeptbücher `source`- und `cookbook`-Deklarationen enthalten.

Wenn AWS OpsWorks Stacks Ihr benutzerdefiniertes Kochbuch-Repository auf einer Instanz installiert, verwendet es Berkshelf, um die abhängigen Kochbücher zu installieren, die im Berksfile des Repositorys deklariert sind. Weitere Informationen finden Sie unter Verwenden von Berkshelf.

Dieses Beispiel zeigt, wie Sie Berkshelf verwenden, um das Community-Kochbuch „Erste Schritte“ auf einer Stacks-Instanz zu installieren. AWS OpsWorks Außerdem installiert Berkshelf eine Version des benutzerdefinierten Rezeptbuchs "createfile", um eine Datei in einem bestimmten Verzeichnis zu installieren. Weitere Informationen zur Funktionsweise von "createfile" finden Sie unter Installieren einer Datei mithilfe eines Rezeptbuchs.

> **Note**
>
> Wenn Sie zum ersten Mal ein benutzerdefiniertes Kochbuch auf einem AWS OpsWorks Stacks-Stack installieren, sollten Sie zuerst das Beispiel durchgehen. Ausführen eines Rezepts auf einer Linux-Instance

Erstellen Sie zunächst einen Stack, wie nachfolgend kurz zusammengefasst. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

Erstellen eines Stacks

1. Öffnen Sie die AWS OpsWorks Stacks-Konsole und klicken Sie auf Add Stack (Stack hinzufügen).
2. Legen Sie die folgenden Einstellungen fest, übernehmen Sie für die restlichen Einstellungen die Standardwerte und klicken Sie auf Add Stack (Stack hinzufügen).

   - Name — BerksTest
   - Standard-SSH-Schlüssel — Ein Amazon EC2 EC2-Schlüsselpaar

   Wenn Sie ein Amazon EC2 EC2-Schlüsselpaar erstellen müssen, finden Sie weitere Informationen unter Amazon EC2 EC2-Schlüsselpaare. Das Schlüsselpaar muss sich in derselben AWS-Region befinden wie die Instance. Das Beispiel verwendet die Standardregion USA West (Oregon).
3. Klicken Sie auf Add a layer (Layer hinzufügen) und fügen Sie dem Stack einen benutzerdefinierten Layer mit folgenden Einstellungen hinzu.

   - Name — BerksTest

- Kurzname — berkstest

  Sie können für dieses Beispiel einen beliebigen Layer-Typ verwenden. Da jedoch für das Beispiel keine Pakete aus den anderen Layers benötigt werden, ist ein benutzerdefinierter Layer die einfachste Lösung.

4. Fügen Sie dem BerksTest Layer eine 24/7-Instanz mit Standardeinstellungen hinzu, aber starten Sie sie noch nicht.

Bei AWS OpsWorks Stacks müssen sich Kochbücher in einem Remote-Repository mit einer Standardverzeichnisstruktur befinden. Anschließend geben Sie die Download-Informationen an AWS OpsWorks Stacks weiter, das das Repository beim Start automatisch auf jede Instanz des Stacks herunterlädt. Der Einfachheit halber ist das Repository für dieses Beispiel ein öffentliches Amazon S3 S3-Archiv, aber AWS OpsWorks Stacks unterstützt auch HTTP-Archive, Git-Repositorys und Subversion-Repositorys. Weitere Informationen finden Sie unter Rezeptbuch-Repositorys.

Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

So erstellen Sie das Rezeptbuch-Repository

1. Erstellen Sie in Ihrem Verzeichnis `opsworks_cookbooks` ein Verzeichnis namens `berkstest_cookbooks`. Sie können dieses Verzeichnis auch an einem beliebigen anderen Ort speichern, da Sie es in ein Repository hochladen werden.
2. Erstellen Sie eine Datei "Berksfile" mit folgendem Inhalt in `berkstest_cookbooks`.

   ```
   source "https://supermarket.chef.io"

   cookbook 'getting-started'
   ```

   Diese Datei enthält die Abhängigkeiten des Rezeptbuchs "getting-started" und weist Berkshelf an, diese auf der Community-Rezeptbuch-Website herunterzuladen.
3. Fügen Sie ein Verzeichnis `createfile` zu `berkstest_cookbooks` hinzu, das Folgendes enthält.

   - Eine Datei `metadata.rb` mit folgendem Inhalt:

```
name "createfile"
version "0.1.0"
```

- Ein Verzeichnis `files/default` mit einer Datei `example_data.json` mit folgendem Inhalt.

```
{
  "my_name" : "myname",
  "your_name" : "yourname",
  "a_number" : 42,
  "a_boolean" : true
}
```

  Sie können den Dateinamen und Inhalt frei wählen. Das Rezept kopiert einfach die Datei an den angegebenen Speicherort.

- Ein Verzeichnis `recipes` mit einer Datei `default.rb` mit folgendem Rezeptcode:

```
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
```

  Dieses Rezept erstellt `/srv/www/shared` und kopiert `example_data.json` in dieses Verzeichnis aus dem Verzeichnis `files` des Rezeptbuchs.

4. Erstellen Sie ein `.zip` Archiv von`berkstest_cookbooks`, laden Sie das Archiv in einen Amazon S3 S3-Bucket hoch, machen Sie das Archiv öffentlich und notieren Sie die URL des Archivs.

Jetzt können Sie die Rezeptbücher installieren und das Rezept ausführen.

So installieren Sie Rezeptbücher und führen die Rezepte aus

1. Bearbeiten Sie den Stack, um benutzerdefinierte Rezeptbücher zu aktivieren, und legen Sie folgende Einstellungen fest:

   - Repository-Typ — HTTP-Archiv
   - Repository-URL — Die URL des Kochbuch-Archivs, die Sie zuvor aufgezeichnet haben
   - Berkshelf verwalten — Ja

   Die ersten beiden Einstellungen versorgen AWS OpsWorks Stacks mit den Informationen, die es benötigt, um das Cookbook-Repository auf Ihre Instanzen herunterzuladen. Die letzte Einstellung aktiviert die Unterstützung für Berkshelf, um das Rezeptbuch "getting-started" auf die Instance herunterzuladen. Übernehmen Sie für die übrigen Einstellungen die Standardwerte und klicken Sie auf Save, um die Stack-Konfiguration zu aktualisieren und zu speichern.
2. Bearbeiten Sie den BerksTest Layer, um die folgenden Rezepte zum Setup-Lifecycle-Ereignis des Layers hinzuzufügen.

   - `getting-started::default`
   - `createfile::default`
3. Starten Sie die Instance. Das Setup-Ereignis tritt ein, nachdem der Startvorgang der Instanz abgeschlossen ist. AWS OpsWorks Stacks installiert dann das Cookbook-Repository, verwendet Berkshelf, um das Kochbuch für die ersten Schritte herunterzuladen, und führt die Einrichtung und Bereitstellung von Rezepten für den Layer aus, einschließlich und. `getting-started::default` `createfile::default`
4. Nachdem die Instance online ist, melden Sie sich mit SSH dort an. Sie sollten Folgendes sehen:

   - `/srv/www/shared` muss `example_data.json` enthalten.
   - `/root` muss `chef-getting-started.txt` enthalten.

     AWS OpsWorks Stacks führt Rezepte als Root-Benutzer aus, sodass Getting-Started die Datei im Verzeichnis und nicht in Ihrem Home-Verzeichnis installiert. `/root`

Verwenden des SDK for Ruby: Dateien von Amazon S3 herunterladen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Es gibt Aufgaben wie die Interaktion mit AWS-Services, die mit Chef-Ressourcen nicht ausgeführt werden können. Es kann beispielsweise in manchen Fällen besser sein, Dateien auf Remote-Servern zu speichern und sie mithilfe von Rezepten auf eine Instance herunterzuladen. Verwenden Sie die Ressource remote_file, um Dateien von Remote-Servern herunterzuladen. Wenn Sie Ihre Dateien jedoch in einem Amazon S3 S3-Bucket speichern möchten, `remote_file` können Sie diese Dateien nur herunterladen, wenn die ACL den Vorgang zulässt.

Rezepte können mithilfe von AWS SDK for Ruby auf die meisten AWS-Services zugreifen. In diesem Thema wird gezeigt, wie Sie das SDK for Ruby verwenden, um eine Datei aus einem S3-Bucket herunterzuladen.

> **Note**
>
> Weitere Informationen zur Verwendung von AWS SDK for Ruby für Verschlüsselung und Entschlüsselung finden Sie unter AWS::S3::S3Object. Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Themen

Verwenden des SDK for Ruby auf einer Vagrant-Instance

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Thema wird beschrieben, wie ein auf einer Vagrant-Instance ausgeführtes Rezept verwendet werden kann AWS SDK for Ruby, um eine Datei von Amazon S3 herunterzuladen. Bevor Sie beginnen, benötigen Sie zunächst eine Reihe von AWS Anmeldeinformationen — einen Zugriffsschlüssel und einen geheimen Zugriffsschlüssel —, die dem Rezept den Zugriff auf Amazon S3 ermöglichen.

> **Important**
>
> Es wird ausdrücklich empfohlen, für diesen Zweck keine Root-Anmeldeinformationen zu verwenden. Erstellen Sie stattdessen einen Benutzer mit einer entsprechenden Richtlinie und geben Sie diese Anmeldeinformationen für das Rezept an.
> Achte darauf, Anmeldeinformationen — auch nicht IAM-Benutzeranmeldedaten — nicht an einem öffentlich zugänglichen Ort zu speichern, indem du beispielsweise eine Datei mit den Anmeldeinformationen in ein öffentliches Repository oder ein Bitbucket-Repository hochlädst. GitHub Dadurch könnten Ihre Anmeldeinformationen offengelegt und die Sicherheit Ihres Kontos beeinträchtigt werden.
> Rezepte, die auf einer EC2Amazon EC2-Instance ausgeführt werden, können einen noch besseren Ansatz verwenden, eine IAM-Rolle, wie unter beschrieben. Verwenden des SDK for Ruby auf einer AWS OpsWorks Stacks-Linux-Instance
> Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Wenn Sie noch keinen geeigneten -Benutzer erstellt haben, erstellen Sie ihn wie nachfolgend beschrieben. Weitere Informationen finden Sie unter Was ist IAM.

> **Warning**
>
> IAM-Benutzer verfügen über langfristige Anmeldeinformationen, was ein Sicherheitsrisiko darstellt. Um dieses Risiko zu minimieren, empfehlen wir, diesen Benutzern nur die Berechtigungen zu gewähren, die sie für die Ausführung der Aufgabe benötigen, und diese Benutzer zu entfernen, wenn sie nicht mehr benötigt werden.

So erstellen Sie einen IAM-Benutzer

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Benutzer und gegebenenfalls Benutzer hinzufügen aus, um einen neuen Administratorbenutzer zu erstellen.
3. Wählen Sie auf der Seite Berechtigungen festlegen die Option Richtlinien direkt anhängen aus.
4. Geben Sie **`S3`** in das Suchfeld Permissions Policies ein, um die Amazon S3 S3-Richtlinien anzuzeigen.

   Wählen Sie AmazonS3. ReadOnlyAccess Wenn Sie möchten, können Sie eine Richtlinie angeben, die umfassendere Berechtigungen gewährt, z. B. AmazonS3 FullAccess. In der Regel werden jedoch nur die erforderlichen Berechtigungen erteilt. In diesem Fall soll das Rezept nur eine Datei herunterladen und benötigt daher nur Lesezugriff.
5. Wählen Sie Weiter aus.
6. Wählen Sie Benutzer erstellen
7. Erstellen Sie als Nächstes Zugangsschlüssel für Ihren Benutzer. Weitere Information über IAM-Zugriffsschlüssel finden Sie unter Verwalten von Zugriffsschlüsseln für IAM-Benutzer im IAM-Benutzerhandbuch.

Nun müssen Sie eine herunterzuladende Datei bereitstellen. In diesem Beispiel wird davon ausgegangen, dass Sie eine Datei `myfile.txt` in einem neu erstellten S3-Bucket `cookbook_bucket` speichern.

So stellen Sie eine Datei zum Herunterladen bereit

1. Erstellen Sie eine Datei `myfile.txt` mit folgendem Text und speichern Sie sie auf Ihrem Computer.

```
This is the file that you just downloaded from Amazon S3.
```

2. Erstellen Sie auf der Amazon S3 S3-Konsole einen Bucket mit dem Namen `cookbook_bucket` in der Region Standard und laden Sie ihn in `myfile.txt` den Bucket hoch.

Richten Sie das Rezeptbuch wie folgt ein.

So richten Sie das Rezeptbuch ein

1. Erstellen Sie ein Verzeichnis in `opsworks_cookbooks` namens `s3bucket` und öffnen Sie es.
2. Initialisieren und konfigurieren Sie Test Kitchen wie unter Beispiel 1: Installieren von Paketen beschrieben.
3. Ersetzen Sie den Text in `.kitchen.yml` durch folgenden.

```
---
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. Fügen Sie zwei Verzeichnisse zu `s3bucket` hinzu: `recipes` und `environments`.
5. Erstellen Sie eine Umgebungsdatei `test.json` mit dem Namen des folgenden `default_attributes` Abschnitts `access_key` und ersetzen Sie dabei die `secret_key`

Werte und durch die entsprechenden Schlüssel für Ihren Benutzer. Speichern Sie die Datei im Verzeichnis `environments` des Rezeptbuchs.

```
{
  "default_attributes" : {
    "cookbooks_101" : {
      "access_key": "AKIAIOSFODNN7EXAMPLE",
      "secret_key" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
    }
  },
  "chef_type" : "environment",
  "json_class" : "Chef::Environment"
}
```

Es gibt mehrere Möglichkeiten, einem Rezept, das auf einer Instance ausgeführt wird, Anmeldeinformationen bereitzustellen. Achten Sie bei der Wahl der richtigen Methode darauf, dass die Schlüssel nicht versehentlich offengelegt werden und somit die Sicherheit Ihres Kontos gefährden. Es wird daher davon ausgeladen, konkrete Schlüsselwerte im Code zu verwenden. In diesem Beispiel werden die Schlüsselwerte stattdessen im Knotenobjekt gespeichert. So kann das Rezept mithilfe der Kontensyntax darauf verweisen, statt die tatsächlichen Werte offenzulegen. Greifen Sie nicht mit Root-Berechtigungen auf das Knotenobjekt zu, um das Risiko, die Schlüssel offenzulegen, möglichst gering zu halten. Weitere Informationen finden Sie unter Bewährte Methoden für die Verwaltung von AWS-Zugriffsschlüsseln.

> **Note**
>
> Im Beispiel werden verschachtelte Attribute mit dem ersten Element `cookbooks_101` verwendet. So sind Namensüberschneidungen unwahrscheinlicher, wenn weitere `access_key`- oder `secret_key`-Attribute im Knotenobjekt vorhanden sind.

Das folgende Rezept lädt `myfile.text` aus dem Bucket `cookbook_bucket` herunter.

```
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
```

Der erste Teil des Rezepts installiert das SDK for Ruby, bei dem es sich um ein Gem-Paket handelt. Die Ressource gem_package installiert Gems, die von Rezepten oder anderen Anwendungen verwendet werden.

> **Note**
>
> Auf Ihrer Instance laufen in der Regel zwei unterschiedliche Versionen von Ruby. Eine davon ist eine Dedicated Instance, die vom Chef-Client verwendet wird. Die andere wird von Anwendungen und Rezepten auf der Instance verwendet. Dies ist ein wichtiger Faktor bei der Installation von Gem-Paketen, da es hierfür zwei Ressourcen gibt, gem_package und chef_gem. Wenn Anwendungen oder Rezepte das das Gem-Paket verwenden, müssen Sie es mit `gem_package` installieren. `chef_gem` ist nur für Gem-Pakete vorgesehen, die vom Chef-Client verwendet werden.

Das restliche Rezept besteht aus einer ruby_block-Ressource, die Ruby-Code zum Herunterladen der Datei enthält. Möglicherweise gehen Sie davon aus, dass Sie den Code direkt in das Rezept schreiben können, da es sich bei dem Rezept ja um eine Ruby-Anwendung handelt. Chef kompiliert den gesamten Code jedoch vor dem Ausführen von Ressourcen. Wenn Sie den Beispielcode direkt im Rezept speichern, versucht Ruby, die `require 'aws-sdk'`-Anweisung aufzulösen, bevor die Ressource `gem_package` ausgeführt wird. Da das SDK for Ruby noch nicht installiert wurde, schlägt die Kompilierung fehl.

Der Code in einer `ruby_block`-Ressource wird hingegen erst dann kompiliert, wenn diese Ressource ausgeführt wird. In diesem Beispiel wird die `ruby_block` Ressource ausgeführt, nachdem die `gem_package` Ressource die Installation des SDK for Ruby abgeschlossen hat, sodass der Code erfolgreich ausgeführt werden kann.

Der Code im `ruby_block` funktioniert folgendermaßen.

1. Er erstellt ein neues `Aws::S3`-Objekt, das die Service-Schnittstelle bereitstellt.

   Der Zugriffsschlüssel und der geheime Schlüssel werden über die im Knotenobjekt gespeicherten Werte referenziert.
2. Er ruft die Verknüpfung `bucket.objects` des `S3`-Objekts auf. Diese gibt ein `Aws::S3::Object`-Objekt mit dem Namen `myfile` zurück, das die Datei `myfile.txt` darstellt.
3. Mithilfe von `Dir.chdir` wird das Arbeitsverzeichnis auf `/tmp` festgelegt.
4. Er öffnet die Datei `myfile.txt`, schreibt den Inhalt von `myfile` in diese Datei und schließt die Datei wieder.

So führen Sie das Rezept aus

1. Erstellen Sie eine Datei `default.rb` mit dem Beispielrezept und speichern Sie sie im Verzeichnis `recipes`.
2. Führen Sie `kitchen converge`.
3. Melden Sie sich mit `kitchen login` bei der Instance an und führen Sie `ls /tmp` aus. Die Datei `myfile.txt` sollte zusammen mit einigen Test Kitchen-Dateien und -Verzeichnissen angezeigt werden.

   ```
   vagrant@s3bucket-ubuntu-1204:~$ ls /tmp
   install.sh  kitchen  myfile.txt  stderr
   ```

   Sie können den Inhalt der Datei auch überprüfen, indem Sie `cat /tmp/myfile.txt` ausführen.

Wenn Sie fertig sind, führen Sie `kitchen destroy` aus, um die Instance zu beenden.

Verwenden des SDK for Ruby auf einer AWS OpsWorks Stacks-Linux-Instance

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Thema wird beschrieben, wie Sie das SDK for Ruby auf einer AWS OpsWorks Stacks-Linux-Instance verwenden, um eine Datei aus einem Amazon S3 S3-Bucket herunterzuladen. AWS OpsWorks Stacks installiert das SDK for Ruby automatisch auf jeder Linux-Instanz. Wenn Sie jedoch das Client-Objekt eines Services erstellen, müssen Sie geeignete AWS-Anmeldeinformationen `AWS::S3.new` oder entsprechende Anmeldeinformationen für andere Services bereitstellen.

Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Verwenden des SDK for Ruby auf einer Vagrant-Instance zeigt, wie Sie Anmeldeinformationen im Knotenobjekt speichern und im Rezeptcode auf die Attribute verweisen, um das Risiko zu minimieren, dass Anmeldeinformationen offengelegt werden. Wenn Sie Rezepte auf einer Amazon EC2 EC2-Instance ausführen, haben Sie eine noch bessere Option, eine IAM-Rolle.

Eine IAM-Rolle funktioniert ähnlich wie ein IAM-Benutzer. Sie verfügen über eine angehängte Richtlinie, die die Berechtigungen für verschiedene AWS-Services enthält. Sie weisen jedoch einer Amazon EC2 EC2-Instance und nicht einer Einzelperson eine Rolle zu. Anwendungen, die auf einer Instance ausgeführt werden, erhalten die Berechtigungen über die angehängte Richtlinie. Bei der Verwendung von Rollen sind die Anmeldeinformationen weder direkt noch indirekt im Code enthalten. In diesem Thema wird beschrieben, wie Sie eine IAM-Rolle verwenden können, um das Rezept Verwenden des SDK for Ruby auf einer Vagrant-Instance auf einer Amazon EC2 EC2-Instance auszuführen.

Sie können dieses Rezept wie in Beispiel 9: Verwenden von Amazon EC2 EC2-Instances beschrieben mit dem kitchen-ec2-Treiber in Test Kitchen ausführen. Die Installation des SDK for Ruby auf Amazon EC2 EC2-Instances ist jedoch etwas kompliziert und nichts, womit Sie sich für AWS OpsWorks Stacks befassen müssen. Auf allen AWS OpsWorks Stacks Linux-Instanzen ist das

SDK for Ruby standardmäßig installiert. Der Einfachheit halber verwendet das Beispiel daher eine AWS OpsWorks Stacks-Instanz.

Der erste Schritt besteht darin, die IAM-Rolle einzurichten. In diesem Beispiel wird der einfachste Ansatz verwendet, nämlich die Amazon EC2 EC2-Rolle zu verwenden, die AWS OpsWorks Stacks erstellt, wenn Sie Ihren ersten Stack erstellen. Sie heißt `aws-opsworks-ec2-role`. AWS OpsWorks Stacks fügt dieser Rolle jedoch keine Richtlinie hinzu und gewährt daher standardmäßig keine Berechtigungen.

Sie müssen die `AmazonS3ReadOnlyAccess` Richtlinie an die `aws-opsworks-ec2-role` Rolle anhängen, um die entsprechenden Berechtigungen zu gewähren. Weitere Informationen zum Anhängen einer Richtlinie an eine Rolle finden Sie unter Hinzufügen von IAM-Identitätsberechtigungen (Konsole) im IAM-Benutzerhandbuch.

Sie legen die Rolle beim Erstellen oder Aktualisieren eines Stacks fest. Richten Sie einen Stack mit einem benutzerdefinierten Layer wie in Ausführen eines Rezepts auf einer Linux-Instance beschrieben ein, allerdings mit einem zusätzlichen Schritt. Vergewissern Sie sich auf der Seite „Stack hinzufügen“, dass das Standard-IAM-Instanzprofil auf 2 Rollen festgelegt ist. aws-opsworks-ec AWS OpsWorks Stacks weist diese Rolle dann allen Instanzen des Stacks zu.

Beim Einrichten des Rezeptbuchs gehen Sie nahezu genauso vor wie unter Ausführen eines Rezepts auf einer Linux-Instance beschrieben. Nachfolgend finden Sie eine kurze Zusammenfassung. Eine ausführliche Erklärung finden Sie im genannten Beispiel.

So richten Sie das Rezeptbuch ein

1. Erstellen Sie ein Verzeichnis `s3bucket_ops` und öffnen Sie es.
2. Erstellen Sie eine Datei `metadata.rb` mit dem folgenden Inhalt und speichern Sie sie unter `s3bucket_ops`.

   ```
   name "s3bucket_ops"
   version "0.1.0"
   ```

3. Erstellen Sie ein Verzeichnis `recipes` in `s3bucket_ops`.
4. Erstellen Sie eine Datei `default.rb` mit dem folgenden Rezept und speichern Sie sie im Verzeichnis `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. Erstellen Sie ein `.zip` Archiv von `s3bucket_ops` und laden Sie das Archiv in einen Amazon S3 S3-Bucket hoch. Der Einfachheit halber veröffentlichen Sie das Archiv und notieren Sie sich die entsprechende URL. Sie können Ihre Kochbücher auch in einem privaten Amazon S3 S3-Archiv oder in verschiedenen anderen Repository-Typen speichern. Weitere Informationen finden Sie unter Rezeptbuch-Repositorys.

Dieses Rezept ist dem im vorherigen Beispiel verwendeten ähnlich, allerdings mit folgenden Ausnahmen.

- Da AWS OpsWorks Stacks das SDK for Ruby bereits installiert hat, wurde die `chef_gem` Ressource gelöscht.
- Das Rezept übergibt keine Anmeldeinformationen an `AWS::S3.new`.

  Die Anmeldeinformationen werden der Anwendung anhand der Rolle der Instance automatisch zugewiesen.
- Das Rezept verwendet `Chef::Log.info`, um dem Chef-Protokoll eine Meldung hinzuzufügen.

Erstellen Sie wie folgt einen Stack für dieses Beispiel. Sie können auch einen vorhandenen Windows-Stack verwenden. Aktualisieren Sie dafür einfach wie nachfolgend beschrieben die Rezeptbücher.

So erstellen Sie einen -Stack

1. Öffnen Sie die AWS OpsWorks Stacks-Konsole und klicken Sie auf Add Stack (Stack hinzufügen).
2. Legen Sie die folgenden Einstellungen fest, übernehmen Sie für die restlichen Einstellungen die Standardwerte und klicken Sie auf Add Stack (Stack hinzufügen).

   - Name — RubySDK
   - Standard-SSH-Schlüssel — Ein Amazon EC2 EC2-Schlüsselpaar

   Wenn Sie ein Amazon EC2 EC2-Schlüsselpaar erstellen müssen, finden Sie weitere Informationen unter Amazon EC2 EC2-Schlüsselpaare. Das Schlüsselpaar muss sich in derselben AWS-Region befinden wie die Instance. Das Beispiel verwendet die Standardregion USA West (Oregon).
3. Klicken Sie auf Add a layer (Layer hinzufügen) und fügen Sie dem Stack einen benutzerdefinierten Layer mit folgenden Einstellungen hinzu.

   - Name — S3Download
   - Kurzname — s3download

   Für Linux-Stacks können Sie einen beliebigen Layer-Typ verwenden. In diesem Beispiel werden jedoch keine der durch die anderen Layer-Typen installierten Pakete benötigt, daher ist es am einfachsten, einen benutzerdefinierten Layer zu verwenden.
4. Fügen Sie dem Layer eine 24/7-Instance mit den Standardeinstellungen hinzu und starten Sie sie.

Jetzt können Sie das Rezept installieren und ausführen.

So führen Sie das Rezept aus

1. Bearbeiten Sie den Stack, um benutzerdefinierte Rezeptbücher zu aktivieren, und legen Sie folgende Einstellungen fest:

   - Repository-Typ — HTTP-Archiv
   - Repository-URL — Die Archiv-URL des Kochbuches, die Sie zuvor aufgenommen haben.

Verwenden Sie für die übrigen Einstellungen die Standardwerte und klicken Sie auf Save (Speichern), um die Stack-Konfiguration zu aktualisieren und zu speichern.

2. Führen Sie den Stack-Befehl „Update Custom Cookbooks" aus, um die aktuelle Version Ihrer benutzerdefinierten Rezeptbücher auf den Stack-Instances zu installieren. Wenn bereits eine ältere Version der Rezeptbücher installiert ist, werden diese überschrieben.
3. Führen Sie das Rezept aus, indem Sie den Stack-Befehl Execute Recipes ausführen. Achten Sie darauf, dass bei Recipes to execute **`s3bucket_ops::default`** eingestellt ist. Durch diesen Befehl wird Chef mit der Option `s3bucket_ops::default` ausgeführt.

> **Note**
>
> In der Regel lassen Sie AWS OpsWorks Stacks Ihre Rezepte automatisch ausführen, indem Sie sie dem entsprechenden Lebenszyklusereignis zuweisen. Sie können diese Rezepte auch durch manuelles Auslösen des Ereignisses ausführen. Verwenden Sie für Einrichtungs- und Konfigurationsereignisse einen Stack-Befehl und für Bereitstellungsereignisse und für Ereignisse zum Aufheben der Bereitstellung einen Bereitstellungsbefehl.

Nachdem das Rezept erfolgreich ausgeführt wurde, können Sie es überprüfen.

So überprüfen Sie s3bucket_ops

1. Werfen Sie zunächst einen Blick in das Chef-Protokoll. Der Stack sollte über eine Instance "opstest1" verfügen. Klicken Sie auf der Seite Instances auf show in der Spalte Log der Instance, um das Chef-Protokoll anzuzeigen. Blättern Sie nach unten zu Ihrem Protokolleintrag.

```
...
[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. Melden Sie sich über SSH bei der Instance an und rufen Sie den Inhalt des Verzeichnisses `/tmp` auf.

Verwenden des SDK for Ruby auf einer AWS OpsWorks Stacks-Windows-Instanz

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> In diesem Beispiel wird davon ausgegangen, dass Sie das Beispiel Ausführen eines Rezepts auf einer Windows-Instance bereits durchgearbeitet haben. Falls Sie das noch nicht getan haben, holen Sie das nun nach. Insbesondere wird darin beschrieben, wie Sie für Ihre Instances RDP-Zugriff aktivieren.
> Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

In diesem Thema wird beschrieben, wie Sie die Windows-Instanz AWS SDK for Rubyauf einer AWS OpsWorks Stacks-Instanz verwenden, um eine Datei aus einem S3-Bucket herunterzuladen.

Wenn eine Ruby-Anwendung Zugriff auf eine AWS-Ressource benötigt, müssen Sie der Anwendung AWS-Anmeldeinformationen mit den entsprechenden Berechtigungen bereitstellen. Für Rezepte ist die Verwendung einer AWS Identity and Access Management (IAM-) Rolle die beste Option für die Bereitstellung von AWS-Anmeldeinformationen. Eine IAM-Rolle funktioniert ähnlich wie ein IAM-Benutzer. Sie hat eine beigefügte Richtlinie, die Berechtigungen zur Nutzung der verschiedenen Dienste gewährt. AWS Sie weisen jedoch einer Amazon Elastic Compute Cloud (Amazon EC2) - Instance statt einer Einzelperson eine Rolle zu. Anwendungen, die auf einer Instance ausgeführt werden, erhalten die Berechtigungen über die angehängte Richtlinie. Bei der Verwendung von Rollen sind die Anmeldeinformationen weder direkt noch indirekt im Code enthalten.

Der erste Schritt besteht darin, die IAM-Rolle einzurichten. In diesem Beispiel wird der einfachste Ansatz verwendet, nämlich die Amazon EC2 EC2-Rolle zu verwenden, die AWS OpsWorks Stacks erstellt, wenn Sie Ihren ersten Stack erstellen. Sie heißt `aws-opsworks-ec2-role`. AWS OpsWorks Stacks fügt dieser Rolle jedoch keine Richtlinie hinzu und gewährt daher standardmäßig keine Berechtigungen.

Sie müssen die `AmazonS3ReadOnlyAccess` Richtlinie an die `aws-opsworks-ec2-role` Rolle anhängen, um die entsprechenden Berechtigungen zu gewähren. Weitere Informationen zum Anhängen einer Richtlinie an eine Rolle finden Sie unter Hinzufügen von IAM-Identitätsberechtigungen (Konsole) im IAM-Benutzerhandbuch.

Sie legen die Rolle beim Erstellen oder Aktualisieren eines Stacks fest. Richten Sie einen Stack mit einem benutzerdefinierten Layer wie in Ausführen eines Rezepts auf einer Windows-Instance beschrieben ein, allerdings mit einem zusätzlichen Schritt. Vergewissern Sie sich auf der Seite „Stack hinzufügen“, dass das Standard-IAM-Instanzprofil auf 2 Rollen festgelegt ist. aws-opsworks-ec AWS OpsWorks Stacks weist diese Rolle dann allen Instanzen des Stacks zu.

Beim Einrichten des Rezeptbuchs gehen Sie nahezu genauso vor wie unter Ausführen eines Rezepts auf einer Linux-Instance beschrieben. Nachfolgend finden Sie eine kurze Zusammenfassung. Eine ausführliche Erklärung finden Sie im genannten Beispiel.

So richten Sie das Rezeptbuch ein

1. Erstellen Sie ein Verzeichnis `s3bucket_ops` und öffnen Sie es.
2. Erstellen Sie eine Datei `metadata.rb` mit dem folgenden Inhalt und speichern Sie sie unter `s3bucket_ops`.

   ```
   name "s3download"
   version "0.1.0"
   ```

3. Erstellen Sie ein Verzeichnis `recipes` in `s3download`.
4. Erstellen Sie eine Datei `default.rb` mit dem folgenden Rezept und speichern Sie sie im Verzeichnis `recipes`. Ersetzen Sie *`windows-cookbooks`* durch den Namen des S3-Buckets, in dem Sie die herunterzuladende Datei speichern möchten.

   ```
   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. Erstellen Sie ein `.zip`-Archiv von `s3download` und laden Sie die Datei in einen S3-Bucket hoch. Machen Sie die Datei öffentlich und notieren Sie sich die URL.
6. Erstellen Sie eine Textdatei `myfile.txt` und laden Sie diese auf einen S3-Bucket hoch. Dies ist die Datei, die Ihr Rezept herunterladen soll, Sie können also einen beliebigen Bucket dafür verwenden.

Das Rezept führt die folgenden Aufgaben aus.

1: Installieren Sie das SDK for Ruby v2.

Das Beispiel verwendet das SDK for Ruby, um das Objekt herunterzuladen. AWS OpsWorks Stacks installiert dieses SDK jedoch nicht auf Windows-Instanzen, sodass der erste Teil des Rezepts eine `chef_gem`Ressource verwendet, um diese Aufgabe zu erledigen. Diese Ressource wird verwendet, um Gems für Chef einschließlich Rezepten zu installieren.

2: Herunterladen der Datei.

Der dritte Teil des Rezepts verwendet eine `ruby_block`Ressource, um SDK for Ruby v2-Code auszuführen, um ihn `myfile.txt` aus einem S3-Bucket herunterzuladen*`windows-cookbooks`*, der in das `/chef` Verzeichnis der Instanz benannt ist. Ändern Sie *`windows-cookbooks`* in den Namen des Buckets, der `myfile.txt` enthält.

> **Note**
>
> Ein Rezept ist eine Ruby-Anwendung. Sie können daher Ruby-Code in den Text des Rezepts kopieren und müssen ihn nicht in einer `ruby_block`-Ressource speichern. Chef führt den Ruby-Code im Text des Rezepts jedoch vor anderen Ressourcen aus. Wenn Sie in diesem Beispiel den Download-Code in den Hauptteil des Rezepts einfügen, schlägt er fehl, da er vom SDK for Ruby abhängt und die `chef_gem` Ressource, die das SDK installiert, noch nicht ausgeführt wurde. Der Code in der `ruby_block` Ressource wird ausgeführt, wenn die Ressource ausgeführt wird, und das geschieht, nachdem die `chef_gem` Ressource das SDK for Ruby installiert hat.

Erstellen Sie wie folgt einen Stack für dieses Beispiel. Sie können auch einen vorhandenen Windows-Stack verwenden. Aktualisieren Sie dafür einfach wie nachfolgend beschrieben die Rezeptbücher.

Erstellen eines Stacks

1. Öffnen Sie die AWS OpsWorks Stacks-Konsole und wählen Sie Add Stack (Stack hinzufügen) aus. Legen Sie die folgenden Einstellungen fest, übernehmen Sie für die restlichen Einstellungen die Standardwerte und wählen Sie Add Stack (Stack hinzufügen) aus.

   - Name — S3Download
   - Region — USA West (Oregon)

     Dieses Beispiel funktioniert in jeder Region, wir empfehlen jedoch, US West (Oregon) für Tutorials zu verwenden.
   - Standardbetriebssystem — Microsoft Windows Server 2012 R2
2. Wählen Sie Add a layer (Layer hinzufügen) aus und fügen Sie dem Stack einen benutzerdefinierten Layer mit folgenden Einstellungen hinzu:

   - Name — S3Download
   - Kurzname — s3download
3. Fügen Sie dem Layer „S3Download” eine 24/7-Instance mit den Standardeinstellungen hinzu und starten Sie sie.

Jetzt können Sie das Rezept installieren und ausführen.

So führen Sie das Rezept aus

1. Bearbeiten Sie den Stack, um benutzerdefinierte Rezeptbücher zu aktivieren, und legen Sie folgende Einstellungen fest:

   - Repository-Typ — S3-Archiv.
   - Repository-URL — Die Archiv-URL des Kochbuches, die Sie zuvor aufgezeichnet haben.

   Übernehmen Sie für die übrigen Einstellungen die Standardwerte und wählen Sie Save aus, um die Stack-Konfiguration zu aktualisieren und zu speichern.
2. Führen Sie den Stack-Befehl „Update Custom Cookbooks" aus, um die aktuelle Version Ihres benutzerdefinierten Rezeptbuchs auf den Online-Instances des Stacks zu installieren. Wenn bereits eine ältere Version der Rezeptbücher installiert ist, werden diese überschrieben.
3. Führen Sie das Rezept aus, indem Sie den Stack-Befehl Execute Recipes ausführen. Achten Sie darauf, dass bei Recipes to execute **`s3download::default`** eingestellt ist. Durch diesen Befehl wird Chef mit der Option `s3download::default` ausgeführt.

   > **Note**
   >
   > In der Regel lassen Sie AWS OpsWorks Stacks Ihre Rezepte automatisch ausführen, indem Sie sie dem entsprechenden Lebenszyklusereignis zuweisen. Sie können diese Rezepte auch durch manuelles Auslösen des Ereignisses ausführen. Verwenden Sie für Einrichtungs- und Konfigurationsereignisse einen Stack-Befehl und für Bereitstellungsereignisse und für Ereignisse zum Aufheben der Bereitstellung einen Bereitstellungsbefehl.

Nachdem das Rezept erfolgreich ausgeführt wurde, können Sie es überprüfen.

So überprüfen Sie s3download

1. Werfen Sie zunächst einen Blick in das Chef-Protokoll. Der Stack sollte über eine Instance "s3download1" verfügen. Wählen Sie auf der Seite Instances die Option show in der Spalte Log der Instance aus, um das Chef-Protokoll anzuzeigen. Blättern Sie nach unten zu Ihrem Protokolleintrag.

   ```
   ...
   ```

```
[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. Melden Sie sich mit RDP bei der Instance an und rufen Sie das Verzeichnis `c:\chef` auf.

Installieren von Windows-Software

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> In diesen Beispielen wird davon ausgegangen, dass Sie das Beispiel Ausführen eines Rezepts auf einer Windows-Instance bereits durchgearbeitet haben. Falls Sie das noch nicht getan haben, holen Sie das nun nach. Insbesondere wird darin beschrieben, wie Sie für Ihre Instances RDP-Zugriff aktivieren.

Auf Windows-Instances ist Windows Server 2012 R2 Standard installiert, daher müssen Sie in der Regel noch einige Softwarepakete installieren. Wie Sie dabei genau vorgehen, hängt von der Art der Software ab.

- Windows-Funktionen sind optionale Systemkomponenten, einschließlich des.NET-Frameworks und Internetinformationsdienste (IIS), die Sie auf Ihre Instanz herunterladen können.
- Drittanbietersoftware verfügt in der Regel über eine Installationsdatei, beispielsweise eine MSI-Datei, die Sie auf die Instance herunterladen und ausführen.

Auch Microsoft-Software verfügt teilweise über ein Installationsprogramm.

In diesem Abschnitt wird beschrieben, wie Sie Rezeptbücher implementieren, um Windows-Funktionen und -Pakete zu installieren. Außerdem wird das Chef-Windows-Rezeptbuch vorgestellt. Dieses enthält Ressourcen und Hilfsfunktionen, die die Rezeptimplementierung auf Windows-Instances vereinfachen.

Themen

- Installieren einer Windows-Funktion: IIS
- Installieren eines Pakets auf einer Windows-Instance

Installieren einer Windows-Funktion: IIS

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Bei den Windows-Funktionen handelt es sich um eine Reihe optionaler Systemkomponenten, einschließlich des.NET-Frameworks und Internetinformationsdienste (IIS). In diesem Thema wird beschrieben, wie Sie ein Kochbuch implementieren, um ein häufig verwendetes Feature, Internetinformationsdienste (IIS), zu installieren.

> **Note**
>
> Installieren eines Pakets zeigt, wie Sie Software mithilfe eines Installationsprogramms, beispielsweise einer MSI-Datei, installieren, die Sie auf die Instance herunterladen und dort ausführen. IIS-Rezeptbücher

In Ausführen eines Rezepts auf einer Windows-Instance wird erläutert, wie Sie mithilfe einer `powershell_script`-Ressource Windows-Funktionen installieren. Dieses Beispiel zeigt

einen alternativen Ansatz: Verwenden Sie die Ressource des Chef Windows-Kochbuchs. `windows_feature` Dieses Rezeptbuch enthält eine Reihe von Ressourcen, die mithilfe von Abbildbereitstellung und Verwaltung (DISM) unterschiedliche Aufgaben, wie die Installation von Funktionen, auf Windows-Systemen ausführen.

> **Note**
>
> Chef verfügt auch über ein IIS-Rezeptbuch, das Sie zur Verwaltung von IIS verwenden können. Weitere Informationen finden Sie unter IIS-Rezeptbuch.

So richten Sie das Rezeptbuch ein

1. Gehen Sie zum GitHub Windows-Kochbuch-Repository und laden Sie das `windows` Kochbuch herunter.

   In diesem Beispiel wird davon ausgegangen, dass Sie das `windows`-Repository als ZIP-Datei herunterladen. Sie können aber auch das Repository klonen.

2. Gehen Sie zum Kochbuch-Repository chef_handler und laden Sie das Kochbuch herunter GitHub . `chef-handler`

   `windows` ist eine Abhängigkeit des Rezeptbuchs `chef_handler` und wird nicht direkt verwendet. In diesem Beispiel wird davon ausgegangen, dass Sie das `chef_handler`-Repository als ZIP-Datei herunterladen. Sie können aber auch das Repository klonen.

3. Entpacken Sie die Rezeptbücher `windows` und `chef_handler` in die Verzeichnisse `windows` und `chef_handler` Ihres Rezeptbuchverzeichnisses.

4. Erstellen Sie ein Unterverzeichnis `install-iis` im Rezeptbuchverzeichnis und öffnen Sie es.

5. Fügen Sie eine Datei `metadata.rb` zu `install-iis` mit dem folgenden Inhalt hinzu:

   ```
   name "install-iis"
   version "0.1.0"

   depends "windows"
   ```

   Mit der Anweisung `depends` können Sie die Ressourcen im Rezeptbuch `windows` in Ihren Rezepten verwenden.

6. Erstellen Sie ein Unterverzeichnis `recipes` in `install-iis` und legen Sie ein Datei `default.rb` mit folgendem Rezeptcode in diesem Verzeichnis an.

```
%w{ IIS-WebServerRole IIS-WebServer }.each do |feature|
  windows_feature feature do
    action :install
  end
end

service 'w3svc' do
  action [:start, :enable]
end
```

   Das Rezept installiert mithilfe der Ressource `windows` des Rezeptbuchs `windows_feature` folgende Komponenten:

   1. Die IIS-Webserver-Rolle
   2. Den IIS-Webserver

   Dann startet und aktiviert das Rezept mithilfe einer `service`-Ressource den IIS-Service (W3SVC).

   > **Note**
   >
   > Um eine vollständige Liste aller verfügbaren Windows-Funktionen anzuzeigen, melden Sie sich mit RDP bei der Instance an, öffnen Sie ein Befehlszeilenfenster und führen Sie den folgenden Befehl aus. Die vollständige Liste ist sehr umfangreich.
   >
   > ```
   > dism /online /Get-Features
   > ```

7. Erstellen Sie ein `.zip`-Archiv, das die Rezeptbücher `install-iis`, `chef_handler` und `windows` enthält und laden Sie das Archiv in einen S3-Bucket hoch. Machen Sie das Archiv öffentlich und notieren Sie sich die URL. In diesem Beispiel wird davon ausgegangen, dass das Archiv den Namen `install-iis.zip` trägt. Weitere Informationen finden Sie unter Rezeptbuch-Repositorys.

Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Erstellen Sie wie folgt einen Stack für dieses Beispiel. Sie können auch einen vorhandenen Windows-Stack verwenden. Aktualisieren Sie dafür einfach wie nachfolgend beschrieben die Rezeptbücher.

Erstellen eines Stacks

1. Öffnen Sie die AWS OpsWorks Stacks-Konsole und wählen Sie Add Stack (Stack hinzufügen) aus. Legen Sie die folgenden Einstellungen fest, übernehmen Sie für die restlichen Einstellungen die Standardwerte und wählen Sie Add Stack (Stack hinzufügen) aus.

   - Name — Installlis
   - Region — USA West (Oregon)

     Dieses Beispiel funktioniert in jeder Region, wir empfehlen jedoch, US West (Oregon) für Tutorials zu verwenden.
   - Standardbetriebssystem — Microsoft Windows Server 2012 R2
2. Wählen Sie Add a layer (Layer hinzufügen) aus und fügen Sie dem Stack einen benutzerdefinierten Layer mit folgenden Einstellungen hinzu:

   - Name — IIS
   - Kurzname — iis
3. Fügen Sie dem IIS-Layer eine 24/7-Instance mit den Standardeinstellungen hinzu und starten Sie sie.

Jetzt können Sie das Rezeptbuch installieren und das Rezept ausführen.

So installieren Sie das Rezeptbuch und führen das Rezept aus

1. Bearbeiten Sie den Stack, um benutzerdefinierte Rezeptbücher zu aktivieren, und legen Sie folgende Einstellungen fest:

   - Repository-Typ — S3-Archiv
   - Repository-URL — Die URL des Kochbucharchivs, die Sie zuvor aufgezeichnet haben.

Übernehmen Sie für die übrigen Einstellungen die Standardwerte und wählen Sie Save aus, um die Stack-Konfiguration zu aktualisieren und zu speichern.

2. Führen Sie den Stack-Befehl Update Custom Cookbooks aus, um die aktuelle Version Ihrer benutzerdefinierten Rezeptbücher auf den Online-Instances des Stacks zu installieren. Wenn bereits eine ältere Version der Rezeptbücher installiert ist, werden diese überschrieben.
3. Führen Sie das Rezept aus, indem Sie den Stack-Befehl Execute Recipes ausführen. Achten Sie darauf, dass bei Recipes to execute **`install-iis::default`** eingestellt ist. Dieser Befehl weist Chef an, die angegebenen Rezepte auszuführen.

> **Note**
>
> In diesem Beispiel wird der Einfachheit halber Execute Recipes verwendet, aber normalerweise lassen Sie AWS OpsWorks Stacks Ihre Rezepte automatisch ausführen, indem Sie sie dem entsprechenden Lebenszyklusereignis zuweisen. Sie können diese Rezepte auch durch manuelles Auslösen des Ereignisses ausführen. Verwenden Sie für Einrichtungs- und Konfigurationsereignisse einen Stack-Befehl und für Bereitstellungsereignisse und für Ereignisse zum Aufheben der Bereitstellung einen Bereitstellungsbefehl.

4. Um die Installation zu überprüfen, melden Sie sich mit RDP bei der Instance an und öffnen Sie den Windows Explorer. Das Dateisystem sollte jetzt über ein Verzeichnis `C:\inetpub` verfügen. IIS sollte in der Systemsteuerung unter Verwaltung in der Liste der Services relativ weit unten aufgeführt sein. Hier trägt es jedoch den Namen World Wide Web Publishing Service und nicht IIS.

Installieren eines Pakets auf einer Windows-Instance

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> In diesem Beispiel wird davon ausgegangen, dass Sie das Beispiel Ausführen eines Rezepts auf einer Windows-Instance bereits durchgearbeitet haben. Falls Sie das noch nicht getan haben, holen Sie das nun nach. Insbesondere wird darin beschrieben, wie Sie für Ihre Instances RDP-Zugriff aktivieren.

Wenn die Software über ein Installationsprogramm wie eine MSI-Datei verfügt, müssen Sie diese Datei auf die Instance herunterladen und dort ausführen. In diesem Beispiel wird gezeigt, wie Sie ein Rezeptbuch implementieren, um ein MSI-Paket, die Python-Laufzeitumgebung, zu installieren und die zugehörigen Umgebungsvariablen zu konfigurieren. Weitere Informationen zum Installieren von Windows-Funktionen wie IIS finden Sie unter Installieren einer Windows-Funktion: IIS.

So richten Sie das Rezeptbuch ein

1. Erstellen Sie ein Verzeichnis `installpython` und öffnen Sie es.
2. Fügen Sie eine Datei `metadata.rb` zu `installpython` mit dem folgenden Inhalt hinzu:

```
name "installpython"
version "0.1.0"
```

3. Fügen Sie die Verzeichnisse `recipes` und `files` zu `installpython` hinzu und fügen Sie ein Verzeichnis `default` zu "files" hinzu.
4. Laden Sie ein Python-Paket für Windows von der Python-Website in das Verzeichnis `files\default` des Rezeptbuchs herunter. In diesem Beispiel wird 3.5.0a3 für Windows x86- mithilfe der MSI-Datei `python-3.4.3.amd64.msi`python-64. installiert.
5. Erstellen Sie im Verzeichnis `default.rb` eine Datei `recipes` mit folgendem Rezeptcode.

```
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
```

Vom Rezept werden folgende Schritte ausgeführt:

1. Es verwendet eine Verzeichnis-Ressource, um ein Verzeichnis `C:\tmp` zu erstellen.

   Weitere Informationen zu dieser Ressource finden Sie unter Beispiel 3: Erstellen von Verzeichnissen.

2. Es verwendet eine cookbook_file-Ressource, um das Installationsprogramm aus dem Verzeichnis `files\default` des Rezeptbuchs in das Verzeichnis `C:\tmp` zu kopieren.

   Weitere Informationen zu dieser Ressource finden Sie unter Installieren einer Datei mithilfe eines Rezeptbuchs.

3. Es verwendet eine windows_package-Ressource, um das MSI-Installationsprogramm auszuführen und Python unter `c:\python34` zu installieren.

   Das Installationsprogramm erstellt die erforderlichen Verzeichnisse und installiert die Dateien. Es nimmt jedoch keine Änderungen an der Umgebungsvariable `PATH` des Systems vor.

4. Es verwendet eine env-Ressource, um `c:\python34` zum Systempfad hinzuzufügen.

   Mit der Ressource "env" werden Umgebungsvariablen festgelegt. In diesem Fall können Sie mit dem Rezept einfach Python-Skripte auf der Befehlszeile ausführen, indem Sie `c:\python34` im Systempfad einfügen.

   - Der Ressourcenname gibt den Namen der Umgebungsvariablen an, in diesem Beispiel `PATH`.

- Über das Attribut `value` wird der Wert der Variablen, in diesem Beispiel `c:\\python34` festgelegt (Sie müssen vor das Zeichen \ "\" setzen).
- Mit der Aktion `:modify` wird der angegebene Wert dem aktuellen Wert der Variablen vorangestellt.
- Das Attribut `delim` setzt ein Trennzeichen, um den neuen Wert von dem vorhandenen Wert zu trennen, in diesem Beispiel `;`.

6. Erstellen Sie ein `.zip`-Archiv von `installpython`, laden Sie das Archiv in einen S3-Bucket hoch und veröffentlichen Sie es. Notieren Sie sich die URL des Archivs. Weitere Informationen finden Sie unter Rezeptbuch-Repositorys.

   Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Erstellen Sie wie folgt einen Stack für dieses Beispiel. Sie können auch einen vorhandenen Windows-Stack verwenden. Aktualisieren Sie dafür einfach wie nachfolgend beschrieben die Rezeptbücher.

Erstellen eines Stacks

1. Öffnen Sie die AWS OpsWorks Stacks-Konsole und wählen Sie Add Stack (Stack hinzufügen) aus. Legen Sie die folgenden Einstellungen fest, übernehmen Sie für die restlichen Einstellungen die Standardwerte und wählen Sie Add Stack (Stack hinzufügen) aus.

   - Name — InstallPython
   - Region — USA West (Oregon)

     Dieses Beispiel funktioniert in jeder Region, wir empfehlen jedoch, US West (Oregon) für Tutorials zu verwenden.
   - Standardbetriebssystem — Microsoft Windows Server 2012 R2

2. Wählen Sie Add a layer (Layer hinzufügen) aus und fügen Sie dem Stack einen benutzerdefinierten Layer mit folgenden Einstellungen hinzu:

   - Bezeichnung — Python
   - Kurzname — Python

3. Fügen Sie dem Python-Layer eine 24/7-Instance mit den Standardeinstellungen hinzu und starten Sie sie.

Nachdem die Instance online ist, können Sie das Rezeptbuch installieren und das Rezept ausführen.

So installieren Sie das Rezeptbuch und führen das Rezept aus

1. Bearbeiten Sie den Stack, um benutzerdefinierte Rezeptbücher zu aktivieren, und legen Sie folgende Einstellungen fest:

   - Repository-Typ — S3-Archiv.
   - Repository-URL — Die Archiv-URL des Kochbuches, die Sie zuvor aufgezeichnet haben.

   Übernehmen Sie für die übrigen Einstellungen die Standardwerte und wählen Sie Save aus, um die Stack-Konfiguration zu aktualisieren und zu speichern.
2. Führen Sie den Stack-Befehl Update Custom Cookbooks aus, um die aktuelle Version Ihrer benutzerdefinierten Rezeptbücher auf den Online-Instances des Stacks zu installieren. Wenn bereits eine ältere Version des Rezeptbuchs installiert ist, wird dieses überschrieben.
3. Führen Sie das Rezept aus, indem Sie den Stack-Befehl Execute Recipes ausführen. Achten Sie darauf, dass bei Recipes to execute **`installpython::default`** eingestellt ist. Durch diesen Befehl wird Chef mit der Option `installpython::default` ausgeführt.

   > **Note**
   >
   > In diesem Beispiel wird der Einfachheit halber Execute Recipes verwendet, aber normalerweise lassen Sie AWS OpsWorks Stacks Ihre Rezepte automatisch ausführen, indem Sie sie dem entsprechenden Lebenszyklusereignis zuweisen. Sie können diese Rezepte auch durch manuelles Auslösen des Ereignisses ausführen. Verwenden Sie für Einrichtungs- und Konfigurationsereignisse einen Stack-Befehl und für Bereitstellungsereignisse und für Ereignisse zum Aufheben der Bereitstellung einen Bereitstellungsbefehl.
4. Um die Installation zu überprüfen, melden Sie sich mit RDP bei der Instance an und öffnen Sie den Windows Explorer.

   - Das Dateisystem sollte jetzt über ein Verzeichnis `C:\Python34` verfügen.
   - Wenn Sie auf der Befehlszeile `path` ausführen, sollte das Ergebnis etwa wie folgt aussehen: `PATH=c:\python34;C:\Windows\system32;...`
   - Wenn Sie auf der Befehlszeile `python --version` ausführen, sollte `Python 3.4.3` zurückgegeben werden.

Überschreiben von integrierten Attributen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Dieses Thema bezieht sich nur auf Linux-Stacks. Auf Windows-Stacks können Sie integrierte Attribute nicht überschreiben.

AWS OpsWorks Stacks installiert auf jeder Instanz eine Reihe integrierter Kochbücher. Viele dieser Rezeptbücher unterstützen die integrierten Layers und über ihre Attributdateien werden zahlreiche Standardsystem- und -anwendungseinstellungen wie die Apache-Serverkonfiguration festgelegt. Wenn Sie diese Einstellungen in Attributdateien speichern, können Sie viele Konfigurationseinstellungen anpassen, indem Sie die entsprechenden integrierten Attribute auf eine der folgenden Weisen überschreiben:

- Definieren Sie das Attribut in benutzerdefinierter JSON.

  Diese Methode ist einfach und flexibel. Allerdings müssen Sie das benutzerdefinierte JSON-Objekt manuell eingeben, daher gibt es keine robuste Lösung, die Attributdefinitionen zu verwalten.
- Implementieren Sie ein benutzerdefiniertes Rezeptbuch und definieren Sie das Attribut in einer Attributdatei `customize.rb`.

  Diese Methode ist zwar weniger flexibel als eine benutzerdefinierte JSON, aber weniger fehleranfällig, da Sie benutzerdefinierte Rezeptbücher an der Quelle kontrollieren können.

In diesem Thema wird anhand des Apache-Servers beispielhaft beschrieben, wie Sie mit der Attributdatei eines benutzerdefinierten Rezeptbuchs integrierte Attribute überschreiben können. Weitere Informationen zum Überschreiben von Attributen mit benutzerdefinierter JSON finden Sie

unter Nutzen eines benutzerdefinierten JSON-Objekts. Eine allgemeine Beschreibung, wie Attribute überschrieben werden, finden Sie unter Überschreiben der Attribute.

> **Note**
>
> Konfigurationseinstellungen lassen sich am besten durch Überschreiben von Attributen anpassen. Jedoch sind Einstellungen nicht immer in Attributen gespeichert. In diesem Fall können Sie die Konfigurationsdatei oft anpassen, indem Sie die Vorlage überschreiben, die von integrierten Rezepten zum Erstellen der Konfigurationsdatei verwendet wird. Ein Beispiel finden Sie unter Überschreiben von integrierten Vorlagen.

Die integrierten Attribute sind in der Regel Werte in den Vorlagendateien, anhand derer Einrichtungsrezepte Konfigurationsdateien erstellen. Zum Beispiel verwendet eines der `apache2`-Einrichtungsrezepte, `default.rb`, die Vorlage `apache2.conf.erb`, um die Hauptkonfigurationsdatei des Apache-Servers, `httpd.conf` (Amazon Linux) oder `apache2.conf` (Ubuntu) zu erstellen. Nachfolgend finden Sie einen Auszug aus der Vorlagendatei:

```
...
#
# 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)
##

...
```

Die Einstellung `KeepAliveTimeout` in diesem Beispiel ist der Wert des Attributs `[:apache][:keepalivetimeout]`. Der Standardwert dieses Attributs wird in der Attributdatei `apache2`apache.rb des Rezeptbuchs festgelegt, wie der nachfolgende Auszug zeigt:

```
...
# 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**
>
> Weitere Informationen zu häufig verwendeten integrierten Attributen finden Sie unter Integrierte Rezeptbuchattribute.

Damit integrierte Attribute überschrieben werden können, enthalten alle integrierten Rezeptbücher die Attributdatei `customize.rb`, die über eine `include_attribute`-Anweisung in allen Modulen integriert ist. Die Datei `customize.rb` eines integrierten Rezeptbuchs enthält keine Attributdefinitionen und wirkt sich nicht auf integrierte Attribute aus. Wenn Sie integrierte Attribute überschreiben möchten, erstellen Sie ein benutzerdefiniertes Rezeptbuch mit demselben Namen wie das integrierte Rezeptbuch und speichern Ihre angepassten Attributdefinitionen in einer Attributdatei mit dem Namen `customize.rb`. Diese Datei hat Vorrang vor der integrierten Version und wird auf allen zugehörigen Modulen gespeichert. Wenn Sie in Ihrer Datei `customize.rb` integrierte Attribute definieren, überschreiben diese die entsprechenden integrierten Attribute.

In diesem Beispiel wird gezeigt, wie Sie das intergierte Attribut `[:apache][:keepalivetimeout]` vom ursprünglichen Wert 3 auf 5 setzen. Dieselbe Methode lässt sich auch auf andere integrierte Attribute anwenden. Achten Sie jedoch darauf, welche Attribute Sie überschreiben. Wenn Sie beispielsweise Attribute im Namespace `opsworks` überschreiben, kann dies zu Problemen mit einigen integrierten Rezepten führen.

> **Important**
>
> Versuchen Sie nicht, integrierte Attribute zu überschreiben, indem Sie eine Kopie der integrierten Attributdatei bearbeiten. Sie könnten zwar eine Kopie von `apache.rb` im Verzeichnis `apache2/attributes` Ihres benutzerdefinierten Rezeptbuchs speichern

und einige Einstellungen anpassen. Diese Datei hat jedoch Vorrang vor der integrierten Version, sodass die integrierten Rezepte nun Ihre Version von `apache.rb` verwenden. Wenn AWS OpsWorks Stacks die integrierte `apache.rb` Datei später ändert, erhalten Rezepte die neuen Werte nicht, es sei denn, Sie aktualisieren Ihre Version manuell. Durch die Verwendung `customize.rb` überschreiben Sie nur die angegebenen Attribute. Die integrierten Rezepte rufen weiterhin automatisch up-to-date Werte für jedes Attribut ab, das Sie nicht überschrieben haben.

Erstellen Sie zunächst ein benutzerdefiniertes Rezeptbuch.

So erstellen Sie das Rezeptbuch

1. Erstellen Sie in Ihrem Verzeichnis `opsworks_cookbooks` ein Rezeptbuchverzeichnis namens `apache2` und öffnen Sie es.

   Damit das benutzerdefinierte Rezeptbuch integrierte Attribute überschreiben kann, muss es denselben Namen wie das integrierte Rezeptbuch haben, in diesem Beispiel also `apache2`.
2. Erstellen Sie im Verzeichnis `apache2` ein Verzeichnis `attributes`.
3. Erstellen Sie eine Datei `customize.rb` im Verzeichnis `attributes` und definieren Sie darin die Attribute des integrierten Rezeptbuchs, die Sie überschreiben möchten. In diesem Beispiel sollte die Datei folgenden Text enthalten:

   ```
   normal[:apache][:keepalivetimeout] = 5
   ```

   > **Important**
   >
   > Damit ein benutzerdefiniertes Attribut ein integriertes Attribut überschreiben kann, muss es mindestens den Typ `normal` sowie denselben Knotennamen wie das entsprechende integrierte Attribut aufweisen. Über den Typ `normal` wird sichergestellt, dass das benutzerdefinierte Attribut Vorrang vor integrierten Attributen hat, die den Typ `default` haben. Weitere Informationen finden Sie unter Priorität von Attributen.
4. Erstellen Sie ein `opsworks_cookbooks` benanntes `.zip` Archiv `opsworks_cookbooks.zip` und laden Sie das Archiv in einen Amazon Simple Storage Service (Amazon S3) -Bucket hoch. Machen Sie die Datei der Einfachheit halber öffentlich. Notieren Sie sich die URL. Sie können

Ihre Kochbücher auch in einem privaten Amazon S3 S3-Archiv oder in anderen Repository-Typen speichern. Weitere Informationen finden Sie unter Rezeptbuch-Repositorys.

Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Erstellen Sie einen Stack und installieren Sie das Rezeptbuch, um das benutzerdefinierte Attribut zu verwenden.

So verwenden Sie benutzerdefinierte Attribute

1. Öffnen Sie die AWS OpsWorks Stacks-Konsole und wählen Sie dann Add Stack (Stack hinzufügen) aus.
2. Legen Sie die folgenden Standardeinstellungen fest.

   - Name — ApacheConfig
   - Region — USA West (Oregon)

     Du kannst deinen Stack in jeder Region platzieren, aber wir empfehlen US West (Oregon) für Tutorials.
   - Standard-SSH-Schlüssel — Ein EC2-Schlüsselpaar

     Falls Sie ein EC2-Schlüsselpaar erstellen müssen, finden Sie dazu weitere Informationen unter Amazon EC2-Schlüsselpaare. Das Schlüsselpaar muss sich in derselben AWS-Region befinden wie der Stack.

   Wählen Sie Advanced **>>** (Erweiterte Einstellungen **>>**) aus, bestätigen Sie die Option Use custom Chef cookbooks (Benutzerdefinierte Rezeptbücher verwenden) mit Yes und legen Sie anschließend die folgenden Einstellungen fest:

   - Repository-Typ — HTTP-Archiv
   - Repository-URL — Die URL des Kochbucharchivs, die Sie zuvor aufgezeichnet haben

   Übernehmen Sie für die anderen Einstellungen die Standardwerte und wählen Sie Add Stack aus, um den Stack zu erstellen.

> **Note**
>
> In diesem Beispiel wird das Standardbetriebssystem Amazon Linux verwendet. Sie können aber auch Ubuntu verwenden. Der einzige Unterschied besteht darin, dass auf Ubuntu-Systemen das integrierte Einrichtungsrezept eine Konfigurationsdatei mit denselben Einstellungen namens `apache2.conf` erstellt und sie im Verzeichnis `/etc/apache2` speichert.

3. Wählen Sie Ebene hinzufügen und fügen Sie dem Stack dann eine Java App Serverebene mit Standardeinstellungen hinzu.
4. Fügen Sie dem Layer eine 24/7-Instance mit den Standardeinstellungen hinzu und starten Sie sie.

   Für dieses Beispiel ist eine t2.Micro-Instance ausreichend.
5. Nachdem die Instance online ist, melden Sie sich mit SSH dort an. Die Datei `httpd.conf` ist im Verzeichnis `/etc/httpd/conf`. In der Datei sehen Sie Ihre benutzerdefinierte Einstellung für `KeepAliveTimeout`. Die übrigen Einstellungen haben die Standardwerte aus der integrierten Datei `apache.rb`. Der relevante Teil der Datei `httpd.conf` sollte etwa wie folgt aussehen:

```
...
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 5
...
```

Überschreiben von integrierten Vorlagen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Dieses Thema bezieht sich nur auf Linux-Stacks. Auf Windows-Stacks können Sie integrierte Vorlagen nicht überschreiben.

Die in AWS OpsWorks Stacks integrierten Rezepte verwenden Vorlagen, um Dateien auf Instanzen zu erstellen, hauptsächlich Konfigurationsdateien für Server wie Apache. Zum Beispiel verwenden die `apache2`-Rezepte die Vorlage `apache2.conf.erb`, um die Hauptkonfigurationsdatei des Apache-Servers zu erstellen, `httpd.conf` (Amazon Linux) oder `apache2.conf` (Ubuntu).

Die meisten Konfigurationseinstellungen in diesen Vorlagen werden durch Attribute abgebildet, daher lassen sich Konfigurationsdateien am besten durch Überschreiben der entsprechenden integrierten Attribute anpassen. Ein Beispiel finden Sie unter Überschreiben von integrierten Attributen. Wenn Sie jedoch Einstellungen anpassen möchten, für die es keine entsprechenden integrierten Attribute gibt oder die in der Vorlage gar nicht vorhanden sind, müssen Sie die Vorlage selbst überschreiben. In diesem Thema wird beschrieben, wie Sie eine integrierte Vorlage überschreiben, um eigene Apache-Konfigurationseinstellungen festzulegen.

Sie können benutzerdefinierte Fehlermeldungen für Apache hinzufügen, indem Sie `ErrorDocument`-Einstellungen in der Datei `httpd.conf` einfügen. `apache2.conf.erb` enthält nur einige auskommentierte Beispiele, wie Sie im Folgenden sehen können:

```
...
#
# 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
...
```

Da diese Einstellungen in Kommentaren fest programmiert sind, können Sie durch Überschreiben der Attribute keine benutzerdefinierten Werte festlegen. Sie müssen die Vorlage selbst überschreiben. Anders als bei Attributen gibt es jedoch keine Möglichkeit, eine Vorlagendatei nur teilweise zu überschreiben. Erstellen Sie ein benutzerdefiniertes Rezeptbuch mit demselben Namen wie die integrierte Version, kopieren Sie die Vorlagendatei in dasselbe Unterverzeichnis und passen Sie die Datei an Ihre Bedürfnisse an. In diesem Thema erfahren Sie, wie Sie die Vorlage `apache2.conf.erb` überschreiben, um für den Fehler 500 eine benutzerdefinierte Fehlermeldung anzuzeigen. Allgemeine Erläuterungen zum Überschreiben von Vorlagen finden Sie unter Verwenden von benutzerdefinierten Vorlagen.

> **Important**
>
> Wenn Sie eine integrierte Vorlage überschreiben, verwenden integrierte Rezepte statt der integrierten Version Ihre angepasste Version der Vorlage. Wenn AWS OpsWorks Stacks die integrierte Vorlage aktualisiert, ist die benutzerdefinierte Vorlage nicht mehr synchron und funktioniert möglicherweise nicht mehr richtig. AWS OpsWorks Stacks nimmt solche Änderungen nicht oft vor, und wenn sich eine Vorlage ändert, listet AWS OpsWorks Stacks die Änderungen auf und gibt Ihnen die Möglichkeit, auf eine neue Version zu aktualisieren. Wir empfehlen Ihnen, auf Änderungen am AWS OpsWorks Stacks-Repository zu achten und Ihre benutzerdefinierte Vorlage gegebenenfalls manuell zu aktualisieren. Das Repository enthält eigene Verzeichnisse für jede unterstützte Chef-Version. Achten Sie daher darauf, das richtige Verzeichnis zu verwenden.

Erstellen Sie zunächst ein benutzerdefiniertes Rezeptbuch.

So erstellen Sie das Rezeptbuch

1. Erstellen Sie in dem Verzeichnis `opsworks_cookbooks` ein Rezeptbuchverzeichnis namens `apache2` und öffnen Sie es anschließend. Damit das benutzerdefinierte Rezeptbuch integrierte Vorlagen überschreiben kann, muss es denselben Namen wie das integrierte Rezeptbuch haben, in diesem Beispiel also `apache2`.

> **Note**
>
> Falls Sie die Anleitung Überschreiben von integrierten Attributen bereits durchgearbeitet haben, können Sie das Rezeptbuch `apache2` aus diesem Beispiel übernehmen und Schritt 2 überspringen.

2. Erstellen Sie eine Datei `metadata.rb` mit folgendem Inhalt und speichern Sie sie im Verzeichnis `apache2`.

```
name "apache2"
version "0.1.0"
```

3. Erstellen Sie im Verzeichnis `apache2` ein Verzeichnis `templates/default`.

> **Note**
>
> Das `templates/default` Verzeichnis funktioniert für Amazon Linux-Instances, die die `apache2.conf.erb` Standardvorlage verwenden. Ubuntu 14.04-Instances verwenden eine für das Betriebssystem spezifische Vorlage `apache2.conf.erb`, die sich im Verzeichnis `templates/ubuntu-14.04` befindet. Wenn Sie Ihre Änderungen auch auf Ubuntu 14.04-Instances verwenden möchten, müssen Sie auch diese Vorlage überschreiben.

4. Kopieren Sie die integrierte Vorlage `apache2.conf.erb` in Ihr Verzeichnis `templates/default`. Öffnen Sie die Vorlagendatei, entfernen Sie die Kommentarzeichen der Zeile `ErrorDocument 500` und geben Sie die folgende benutzerdefinierte Fehlermeldung ein:

```
...
ErrorDocument 500 "A custom error message."
#ErrorDocument 404 /missing.html
...
```

5. Erstellen Sie ein `.zip` Archiv `opsworks_cookbooks` mit Namen `opsworks_cookbooks.zip` und laden Sie die Datei dann in einen Amazon Simple Storage Service (Amazon S3) -Bucket hoch. Machen Sie das Archiv der Einfachheit halber öffentlich. Notieren Sie sich die URL des Archivs. Sie können Ihre Kochbücher auch in einem privaten Amazon S3 S3-Archiv oder in

anderen Repository-Typen speichern. Weitere Informationen finden Sie unter Rezeptbuch-Repositorys.

Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

> **Note**
>
> Der Einfachheit halber wird in diesem Beispiel eine fest programmierte Meldung in die Vorlage eingefügt. Um diese zu ändern, müssen Sie die Vorlage ändern und das Rezeptbuch erneut installieren. Wenn Sie flexibler sein möchten, können Sie ein benutzerdefiniertes Standardattribut für die Fehlermeldung in der Attributdatei des benutzerdefinierten Rezeptbuchs `customize.rb` definieren und den Wert dieses Attributs `ErrorDocument 500` zuweisen. Wenn Sie das Attribut beispielsweise `[:apache][:custom][:error500]` nennen, sieht die entsprechende Zeile in der Datei `apache2.conf.erb` etwa folgendermaßen aus:
>
> ```
> ...
> ErrorDocument 500 <%= node[:apache][:custom][:error500] %>
> #ErrorDocument 404 /missing.html
> ...
> ```
>
> Nun können Sie die benutzerdefinierte Fehlermeldung jederzeit ändern, indem Sie `[:apache][:custom][:error500]` überschreiben. Wenn Sie das Attribut mit benutzerdefinierter JSON überschreiben, müssen Sie das Rezeptbuch überhaupt nicht bearbeiten.

Erstellen Sie einen Stack und installieren Sie das Rezeptbuch, um die benutzerdefinierte Vorlage zu verwenden.

So verwenden Sie benutzerdefinierte Vorlagen

1. Öffnen Sie die AWS OpsWorks Stacks-Konsole und wählen Sie dann Add Stack (Stack hinzufügen) aus.
2. Legen Sie die folgenden Standardeinstellungen fest:

- Name — ApacheTemplate
- Region — USA West (Oregon)
- Standard-SSH-Schlüssel — Ein Amazon Elastic Compute Cloud (Amazon EC2) - Schlüsselpaar

  Wenn Sie ein Amazon EC2 EC2-Schlüsselpaar erstellen müssen, finden Sie weitere Informationen unter Amazon EC2 EC2-Schlüsselpaare. Das Schlüsselpaar muss sich in derselben AWS-Region befinden wie die Instance.

Wählen Sie Advanced **>>** (Erweiterte Einstellungen **>>**) und dann Use custom Chef cookbooks (Benutzerdefinierte Rezeptbücher verwenden) aus, um die folgenden Einstellungen anzugeben:

- Repository-Typ — HTTP-Archiv
- Repository-URL — Die URL des Kochbucharchivs, die Sie zuvor aufgezeichnet haben

Übernehmen Sie für die anderen Einstellungen die Standardwerte und wählen Sie Add Stack aus, um den Stack zu erstellen.

3. Wählen Sie Ebene hinzufügen und fügen Sie dem Stack dann eine Java App Server-Ebene mit Standardeinstellungen hinzu.
4. Fügen Sie dem Layer eine 24/7-Instance mit den Standardeinstellungen hinzu und starten Sie sie.

   Für dieses Beispiel ist eine t2.Micro-Instance ausreichend.
5. Nachdem die Instance online ist, melden Sie sich mit SSH dort an. Die Datei `httpd.conf` ist im Verzeichnis `/etc/httpd/conf`. Die Datei sollte nun Ihre benutzerdefinierte Einstellung für `ErrorDocument` enthalten, die etwa folgendermaßen aussieht:

```
...
# 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
...
```

Load Balancing eines Layers

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks bietet zwei Load-Balancing-Optionen, Elastic Load Balancing und HAProxy, die in der Regel verwendet werden, um die Last auf die Instanzen einer Anwendungsserverebene zu verteilen. In diesem Thema werden die Vorteile und Einschränkungen der beiden Optionen beschrieben, sodass Sie besser entscheiden können, welche Option sich für Sie besser eignet, wenn Sie eine Lastverteilungsfunktion zu einem Layer hinzufügen möchten. In einigen Fällen ist es am besten, beide Optionen zu verwenden.

SSL-Terminierung

Die integrierte HAProxy-Schicht verarbeitet keine SSL-Terminierung. Sie müssen SSL auf den Servern beenden. Das hat den Vorteil, dass der Datenverkehr verschlüsselt ist, bis er die Server erreicht. Allerdings müssen die Server die Verschlüsselung verarbeiten, wodurch deren Last erhöht wird. Darüber hinaus müssen Sie Ihre SSL-Zertifikate auf den Anwendungsservern speichern, auf die Benutzer leichter zugreifen können.

Mit Elastic Load Balancing können Sie SSL am Load Balancer beenden. Dadurch wird die Belastung Ihrer Anwendungsserver reduziert, der Datenverkehr zwischen dem Load Balancer und dem Server wird jedoch nicht verschlüsselt. Elastic Load Balancing ermöglicht es Ihnen auch, SSL auf dem Server zu beenden, aber die Einrichtung ist etwas kompliziert.

Skalierung

Wenn der eingehende Datenverkehr die Kapazität eines HAProxy-Load Balancers übersteigt, müssen Sie diese manuell erhöhen.

Elastic Load Balancing skaliert automatisch, um eingehenden Datenverkehr zu verarbeiten. Um sicherzustellen, dass ein Elastic Load Balancing Load Balancer über ausreichend Kapazität verfügt, um die zu erwartende Last zu bewältigen, wenn er zum ersten Mal online geht, können Sie ihn vorwärmen.

Load Balancer-Ausfall

Wenn die Instance ausfällt, auf der Ihr HAProxy-Server gehostet wird, ist möglicherweise die gesamte Website so lange offline, bis Sie die Instance neu starten können.

Elastic Load Balancing ist ausfallresistenter als HAProxy. So werden beispielsweise Load Balancing-Knoten für jede verfügbare Availability Zone bereitgestellt, für die EC2 Instances registriert wurden. Wenn der Service in einer Zone unterbrochen wird, können die anderen Knoten weiterhin den eingehenden Datenverkehr verarbeiten. Weitere Informationen finden Sie unter Elastic Load Balancing Concepts.

Timeout bei Leerlauf

Beide Load Balancer beenden eine Verbindung, wenn sich ein Server für eine bestimmte Zeit im Leerlauf befindet.

- HAProxy — Der Wert für das Leerlauf-Timeout hat keine Obergrenze.
- Elastic Load Balancing — Der Standardwert für das Leerlauf-Timeout beträgt 60 Sekunden mit einem Maximum von 3600 Sekunden (60 Minuten).

Das Leerlaufzeitlimit von Elastic Load Balancing ist für die meisten Zwecke ausreichend. Wir empfehlen die Verwendung von HAProxy, wenn eine längere Leerlaufzeit benötigt wird. Beispielsweise:

- Eine lange andauernde HTTP-Verbindung für Push-Benachrichtigungen
- Eine administrative Schnittstelle, mit der Sie Aufgaben ausführen, die länger als 60 Minuten dauern

URL-basierte Zuweisung

Sie können einen Load Balancer anweisen, eine eingehende Anforderung an einen bestimmten Server basierend auf der URL der Anforderung zu übermitteln. Angenommen, Sie haben eine Gruppe von zehn Anwendungsservern, die eine kommerzielle Online-Anwendung unterstützen. Acht der Server verarbeiten den Katalog und zwei die Zahlungen. Sie möchte alle HTTP-Anforderungen, die mit der Zahlung zusammenhängen, basierend auf der Anforderungs-URL an die Zahlungsserver umleiten. In diesem Fall würden Sie alle URLs, die „Zahlung„ oder „Auschecken“ beinhalten, an einen der Zahlungsserver umleiten.

Mit HAProxy können Sie über die URL-basierte Zuweisung URLs mit bestimmten Zeichenfolgen an bestimmte Server umleiten. Um das URL-basierte Mapping mit AWS OpsWorks Stacks zu verwenden, müssen Sie eine benutzerdefinierte HAProxy-Konfigurationsdatei erstellen, indem Sie die `haproxy-default.erb` Vorlage im integrierten Cookbook überschreiben. `haproxy`

Weitere Informationen finden Sie im HAProxy Configuration Manual und unter Verwenden von benutzerdefinierten Vorlagen. Sie können URL-basierte Zuweisungen nicht für HTTPS-Anforderungen nutzen. Eine HTTPS-Anforderung ist verschlüsselt, somit kann HAProxy die Anforderungs-URL nicht überprüfen.

Elastic Load Balancing bietet eingeschränkte Unterstützung für URL-Mapping. Weitere Informationen finden Sie unter Listener Configurations for Elastic Load Balancing (Listener-Konfigurationen für Elastic Load Balancing).

Empfehlung: Wir empfehlen die Verwendung von Elastic Load Balancing für den Load Balancing, sofern Sie keine Anforderungen haben, die nur von HAProxy erfüllt werden können. In diesem Fall könnte der beste Ansatz darin bestehen, beide zu kombinieren, indem Elastic Load Balancing als Frontend-Load Balancer verwendet wird, der den eingehenden Traffic auf eine Reihe von HAProxy-Servern verteilt. So gehen Sie vor:

- Richten Sie eine HAProxy-Instance in den einzelnen Availability Zones des Stacks ein, um Anforderungen auf die Anwendungsserver der Zone zu verteilen.
- Weisen Sie die HAProxy-Instances einem Elastic Load Balancing Load Balancer zu, der dann eingehende Anfragen an die HAProxy Load Balancer verteilt.

Dieser Ansatz ermöglicht es Ihnen, die URL-basierte Zuweisung von HAProxy zur Verteilung unterschiedlicher Arten von Anforderungen an die entsprechenden Anwendungsserver zu nutzen. Wenn jedoch einer der HAProxy-Server offline geht, funktioniert die Site weiterhin, da der Elastic Load Balancing Load Balancer den eingehenden Traffic automatisch auf die fehlerfreien HAProxy-Server verteilt. Beachten Sie, dass Sie Elastic Load Balancing als Frontend-Load Balancer verwenden müssen. Ein HAProxy-Server kann Anfragen nicht an andere HAProxy-Server verteilen.

Migration von Chef Server zu Stacks AWS OpsWorks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Da AWS OpsWorks Stacks auf Chef basiert, ist die Migration von Chef Server zu AWS OpsWorks Stacks relativ einfach. Dieser Abschnitt enthält Leitlinien zum Anpassen von Chef-Server-Code auf AWS OpsWorks Stacks.

> **Note**
>
> Wir raten davon ab, eine Migration auf Stacks mit Chef-Versionen niedriger als 11.10, die auf Chef-Solo basieren und keine Suche oder Data Bags unterstützen, durchzuführen.

Themen

Zuweisen von Rollen an Layer

Chef-Server verwendet Rollen für die Darstellung und Verwaltung von Instances mit demselben Zweck und derselben Konfiguration, wie zum Beispiel eine Gruppe von Instances, die jeweils einen Java-Anwendungsserver hosten. Eine AWS OpsWorks Stacks-Ebene erfüllt im Wesentlichen denselben Zweck wie eine Chef-Rolle. Eine Ebene ist eine Blaupause für die Erstellung einer Reihe von Amazon Elastic Compute Cloud (Amazon EC2) -Instances mit derselben Konfiguration, denselben installierten Paketen, demselben Anwendungsbereitstellungsverfahren usw.

AWS OpsWorks Stacks umfasst eine Reihe integrierter Ebenen für verschiedene Arten von Anwendungsservern, einen HAProxy-Loadbalancer, einen MySQL-Datenbankmaster und einen Ganglia-Monitoring-Master. Die integrierte Java App Server-Ebene ist beispielsweise eine Blaupause für die Erstellung von Instanzen, die einen Tomcat-Server hosten.

Um zu AWS OpsWorks Stacks zu migrieren, müssen Sie jede Rolle einer Ebene zuordnen, die entsprechende Funktionen bietet. Für einige Rollen können Sie einfach eines der integrierten Layer verwenden. Für andere Rollen ist ggf. eine mehr oder weniger umfangreiche Anpassung erforderlich. Beginnen Sie mit der Prüfung der Funktionalität der integrierten Layer, einschließlich der jeweils zugeordneten Rezepte, um festzustellen, ob mindestens einer über die Funktionalität Ihrer Rolle verfügt. Weitere Informationen zu den integrierten Layern finden Sie unter Ebenen und

AWS OpsWorks Stacks-Ebenenreferenz. Informationen zu den integrierten Rezepten finden Sie im öffentlichen AWS OpsWorks GitHub Stacks-Repository.

Die Vorgehensweise hängt davon ab, wie gut Sie ein Layer für die jeweilige Rolle anpassen können.

Ein integrierter Layer unterstützt sämtliche Funktionen der Rolle

Sie können den integrierten Layer direkt, ggf. mit geringfügigen Anpassungen, verwenden. Wenn eine Rolle beispielsweise einen Tomcat-Server unterstützt, übernehmen die Rezepte der Java App Server-Schicht möglicherweise bereits alle Aufgaben der Rolle, möglicherweise mit einigen geringfügigen Anpassungen. Sie können beispielsweise veranlassen, dass die integrierten Rezepte der Ebene Tomcat- oder Apache-Konfigurationseinstellungen verwenden, indem die entsprechenden Attribute oder Vorlagen überschrieben werden.

Ein integrierter Layer unterstützt einige, aber nicht alle Funktionalitäten einer Rolle

Sie können ggf. einen integrierte Ebene mithilfe einer Ebenenerweiterung verwenden. Dazu ist normalerweise die Implementierung von benutzerdefinierten Rezepten zur Unterstützung der fehlenden Funktionalität und die Zuweisung der Rezepte auf die Lebenszyklusereignisse des Layers erforderlich. Angenommen, Ihre Rolle installiert einen Redis-Server auf derselben Instance, die einen Tomcat-Server hostet. Sie könnten die Java App Server-Ebene so erweitern, dass sie der Funktionalität der Rolle entspricht, indem Sie ein benutzerdefiniertes Rezept für die Installation von Redis auf den Instanzen der Ebene implementieren und das Rezept dem Setup-Ereignis der Ebene zuweisen.

Die Funktionalität der Rolle wird von keinem integrierten Layer ausreichend unterstützt

Implementieren Sie einen benutzerdefinierten Layer. Angenommen, Ihre Rolle unterstützt einen MongoDB-Datenbank-Server, der von keinem der integrierten Layer unterstützt wird. Sie können diese Unterstützung herstellen, indem Sie die Rezepte implementieren, um die erforderlichen Pakete zu installieren, den Server zu konfigurieren usw. und um die Rezepte einem Lebenszyklusereignis eines benutzerdefinierten Layers zuzuweisen. In der Regel können Sie hierfür mindestens einige der Rezepte der Rolle verwenden. Weitere Informationen zum Implementieren eines benutzerdefinierten Layers finden Sie unter Erstellen eines benutzerdefinierten Tomcat-Server-Layers.

Verwenden von Data Bags

Chef-Server ermöglicht die Übermittlung von benutzerdefinierten Daten an Ihre Rezepte mithilfe von Data Bags.

- Speichern Sie die Daten mit Ihren Rezeptbüchern und Chef installiert diese auf jeder Instance.
- Sie können verschlüsselte Data Bags für vertrauliche Daten verwenden, wie z. B. Passwörter.

AWS OpsWorks Stacks unterstützt Datenbeutel; Rezepte können die Daten mit genau demselben Code wie bei Chef Server abrufen. Die Unterstützung hat jedoch folgende Einschränkungen und Unterschiede:

- Data Bags werden nur von Chef 11.10 Linux und höheren Stacks unterstützt.

  Windows-Stacks und Linux-Stacks unter niedrigeren Versionen von Chef unterstützen Data Bags nicht.
- Speichern Sie keine Data Bags in Ihrem Rezeptbuch-Repository.

  Verwenden Sie stattdessen ein benutzerdefiniertes JSON-Objekt zur Verwaltung der Daten des Data Bags.
- AWS OpsWorks Stacks unterstützt keine verschlüsselten Datenbeutel.

  Wenn Sie vertrauliche Daten in verschlüsselter Form, wie z. B. Passwörter oder Zertifikate, speichern müssen, empfehlen wir, diese in einem privaten S3-Bucket zu speichern. Anschließend können Sie ein benutzerdefiniertes Rezept erstellen, das zum Abrufen der Daten das Amazon SDK für Ruby verwendet. Ein Beispiel finden Sie unter Verwenden des -SDK for Ruby.

Weitere Informationen finden Sie unter Verwenden von Data Bags.

Verwenden der Chef-Suchfunktion

Chef-Server speichert Stack-Konfigurationsinformationen wie IP-Adressen und Rollen-Konfigurationen auf dem Server. Rezepte verwenden die Chef-Suche, um diese Daten abzurufen. AWS OpsWorks Stacks verwendet einen etwas anderen Ansatz. Zum Beispiel basieren Chef 11.10 Linux-Stacks auf dem Chef-Client-Lokal-Modus, eine Chef-Client-Option, die eine abgespeckte Chef-Server-Version (oft als Chef Zero bezeichnet) lokal auf der Instance ausführt. Chef Zero unterstützt die Suche von Daten, die im Knotenobjekt der Instance gespeichert sind.

Anstatt Stack-Daten auf einem Remote-Server zu speichern, fügt AWS OpsWorks Stacks dem Knotenobjekt jeder Instanz für jedes Lebenszyklusereignis eine Reihe von Stackkonfigurations- und Bereitstellungsattributen hinzu. Diese Attribute stellen einen Snapshot der Stack-Konfiguration dar. Sie verwenden dieselbe Syntax wie der Chef-Server und enthalten die meisten Daten, die von den Rezepten vom Server abgerufen werden müssen.

Oft müssen Sie den suchabhängigen Code Ihrer Rezepte für Stacks nicht ändern. AWS OpsWorks Da die Chef-Suche auf dem Knotenobjekt basiert, das die Stackkonfiguration und die Bereitstellungsattribute enthält, funktionieren Suchanfragen in AWS OpsWorks Stacks normalerweise genauso wie bei Chef Server.

Die Hauptausnahme wird durch die Tatsache verursacht, dass die Stack-Konfiguration und die Bereitstellungsattribute nur Daten enthalten, die AWS OpsWorks Stacks bei der Installation der Attribute auf der Instanz bekannt sind. Wenn Sie ein Attribut lokal auf einer bestimmten Instance erstellen oder ändern, werden diese Änderungen nicht an AWS OpsWorks Stacks weitergegeben und werden nicht in die Stack-Konfiguration und die Bereitstellungsattribute übernommen, die auf den anderen Instances installiert sind. Sie können die Suchfunktion nur zum Abrufen eines Attributwertes auf dieser Instance verwenden. Weitere Informationen finden Sie unter Verwenden der Chef-Suchfunktion.

Aus Gründen der Kompatibilität mit Chef Server fügt AWS OpsWorks Stacks dem Knotenobjekt eine Reihe von `role` Attributen hinzu, von denen jedes eines der Layer-Attribute des Stacks enthält. Wenn Ihr Rezept `roles` als Such-Schlüssel verwendet, müssen Sie den Such-Code nicht ändern. Die Abfrage gibt automatisch die Daten für den entsprechenden Layer zurück. Die beiden folgenden Abfragen geben beispielsweise die `php-app`-Attribute des Layers zurück.

```
phpserver = search(:node, "layers:php-app").first
```

```
phpserver = search(:node, "roles:php-app").first
```

Verwalten von Rezeptbüchern und Rezepten

AWS OpsWorks Stacks und Chef Server behandeln Kochbücher und Rezepte etwas anders. Chef Server:

- Sie stellen alle Rezeptbücher zur Verfügung, indem Sie sie entweder selbst oder mithilfe von Community-Rezeptbüchern implementieren.
- Speichern Sie Ihre Rezeptbücher auf dem Server.
- Führen Sie die Rezepte manuell oder planmäßig aus.

Mit AWS OpsWorks Stacks:

- AWS OpsWorks Stacks bietet ein oder mehrere Kochbücher für jede der integrierten Ebenen. Diese Rezeptbücher verarbeiten Standardaufgaben, wie z. B. das Installieren und Konfigurieren einer integrierten Layer-Software und Bereitstellen von Anwendungen.

  Zum Verarbeiten von Aufgaben, die nicht von den integrierten Rezeptbüchern durchgeführt werden, fügen Sie benutzerdefinierte Rezeptbücher zu Ihrem Stack hinzu oder verwenden Sie Community-Rezeptbücher.
- Sie speichern AWS OpsWorks Stacks-Kochbücher in einem Remote-Repository, z. B. in einem S3-Bucket oder einem Git-Repository.

  Weitere Informationen finden Sie unter Speichern von Rezeptbüchern.
- Sie können Rezepte manuell ausführen, aber normalerweise lassen Sie AWS OpsWorks Stacks Rezepte für Sie als Reaktion auf eine Reihe von Lebenszyklusereignissen ausführen, die an wichtigen Punkten im Lebenszyklus einer Instanz auftreten.

  Weitere Informationen finden Sie unter Ausführen von Rezepten.
- AWS OpsWorks Stacks unterstützt Berkshelf nur auf Chef 11.10-Stacks. Wenn Sie die Rezeptbuch-Abhängigkeiten mit Berkshelf verwalten, können Sie keine Stacks mit Chef 11.4 oder niedriger verwenden.

  Weitere Informationen finden Sie unter Verwenden von Berkshelf.

Themen

Speichern von Rezeptbüchern

Mit dem Chef-Server speichern Sie Ihre Rezeptbücher auf dem Server und stellen Sie vom Server für die Instances bereit. Mit AWS OpsWorks Stacks speichern Sie Kochbücher in einem Repository — einem S3- oder HTTP-Archiv oder einem Git- oder Subversion-Repository. Sie geben die Informationen an, die AWS OpsWorks Stacks benötigt, um den Code aus dem Repository auf die Instanzen eines Stacks herunterzuladen, wenn Sie Cookbooks installieren.

Für die Migration vom Chef-Server müssen Sie Ihre Rezeptbücher in einem dieser Repositorys ablegen. Weitere Informationen zur Struktur eines Rezeptbuch-Repositorys finden Sie unter Rezeptbuch-Repositorys.

Ausführen von Rezepten

In AWS OpsWorks Stacks hat jede Ebene eine Reihe von Lebenszyklusereignissen — Setup, Configure, Deploy, Undeploy und Shutdown —, die jeweils an einem wichtigen Punkt im Lebenszyklus einer Instanz auftreten. Um ein benutzerdefiniertes Rezept auszuführen, weisen Sie es in der Regel dem entsprechenden Ereignis auf dem zugehörigen Layer zu. Wenn das Ereignis eintritt, führt AWS OpsWorks Stacks die entsprechenden Rezepte aus. Beispielsweise findet das Einrichtungsereignis nach Abschluss eines Bootvorgangs einer Instance statt. Daher weisen Sie normalerweise diesem Ereignis Rezepte zu, die Aufgaben wie das Installieren und Konfigurieren von Paketen und Starten von Services ausführen.

Sie können Rezepte mit dem Stack-Befehl "Execute Recipes" ausführen. Dieser Befehl ist zum Entwickeln und Testen hilfreich, aber Sie können ihn auch zum Ausführen von Rezepten verwenden, die nicht zu einem Lebenszyklusereignis zugewiesen sind. Sie können auch den Befehl zum Ausführen von Rezepten verwenden, um Einrichtungs- und Konfigurationsereignisse manuell auszulösen.

Zusätzlich zur AWS OpsWorks Stacks-Konsole können Sie die AWS-CLI oder SDKs verwenden, um Rezepte auszuführen. Diese Tools unterstützen sämtliche AWS OpsWorks Stacks-API-Aktionen, sind aber einfacher zu verwenden als die API. Verwenden Sie den CLI-Befehl „create-deployment“, um ein Lebenszyklusereignis auszulösen, das alle zugeordneten Rezepte ausführt. Mit diesem Befehl können Sie auch eine oder mehrere Rezepte ausführen, ohne ein Ereignis auszulösen. Der entsprechende SDK-Code hängt von der spezifischen Sprache ab, ist aber in der Regel dem CLI-Befehl ähnlich.

Die folgenden Beispiele beschreiben zwei Möglichkeiten zur Nutzung des CLI-Befehls `create-deployment`, um die Anwendungsbereitstellung zu automatisieren.

- Stellen Sie Ihre App planmäßig bereit, indem Sie einen benutzerdefinierten Layer mit einer einzelnen Instance auf Ihrem Stack hinzufügen.

  Fügen Sie ein benutzerdefiniertes Einrichtungsrezept hinzu, das einen `cron`-Auftrag in der Instance erstellt, um den Befehl nach einem bestimmten Zeitplan auszuführen. Ein Beispiel für die Verwendung eines Rezepts zum Erstellen eines `cron`-Auftrags finden Sie unter Ausführen von Cron-Jobs auf Linux-Instances.
- Fügen Sie zur Ihrer laufenden Integrationspipeline eine Aufgabe hinzu, die den CLI-Befehl `create-deployment` zur Bereitstellung der Anwendung verwendet.

Verwenden von Chef-Umgebungen

AWS OpsWorks Stacks unterstützt keine Chef-Umgebungen; `node.chef_environment` kehrt immer zurück. `_default`

AWS OpsWorks Stacks-Ebenenreferenz

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Jede Instanz, die AWS OpsWorks Stacks bereitstellt, muss mindestens einer Ebene angehören, die die Rolle einer Instanz im Stack definiert und die Einzelheiten der Einrichtung und Konfiguration der Instanz, der Installation von Paketen, der Bereitstellung von Anwendungen usw. steuert. Weitere Informationen zur Verwendung der AWS OpsWorks Stacks zum Erstellen und Verwalten von Ebenen finden Sie unter. Ebenen

Jede Layer-Beschreibung enthält eine Liste der integrierten Rezepte, die AWS OpsWorks Stacks für jedes Lifecycle-Ereignis der Ebene ausführt. Diese Rezepte sind unter https://github.com/aws/opsworks-cookbooks gespeichert. Beachten Sie, dass die Listen nur die Rezepte enthalten, die direkt von AWS OpsWorks Stacks ausgeführt werden. Diese Rezepte führen manchmal abhängige Rezepte aus, die nicht aufgeführt sind. Wenn Sie für ein bestimmtes Ereignis die vollständige Liste der Rezepte sehen möchten, einschließlich der abhängigen und benutzerdefinierten Rezepte, prüfen Sie die Ausführungsliste im entsprechenden Chef-Protokoll des Lebenszyklusereignisses.

Themen

- HAProxy Layer-Referenz
- HAProxy Stacks AWS OpsWorks , Ebene
- MySQL-Ebenenreferenz
- OpsWorks MySQL-Schicht
- Referenz für Anwendungsserver-Layer
- Anwendungsserverebene

- Referenz zur ECS-Clusterschicht
- Referenz für benutzerdefinierte Layer
- Referenz für andere Layer
- Andere Layer

HAProxy Layer-Referenz

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Eine HAProxy-Schicht verwendet HAProxy — einen zuverlässigen Hochleistungs-TCP/HTTP-Loadbalancer —, um hochverfügbare Lastenausgleichs- und Proxydienste für TCP- und HTTP-basierte Anwendungen bereitzustellen. Das ist besonders nützlich für Websites, die unter sehr hohen Belastungen durchsucht werden müssen und gleichzeitig Persistenz- oder Layer 7-Verarbeitung erfordern.

HAProxy überwacht den Datenverkehr und zeigt die Statistiken und den Zustand der zugeordneten Instances auf einer Webseite an. *`Standardmäßig lautet der URI http://dnsName /haproxy?`* stats, wobei *`dnsName der DNS-Name der HAProxy-Instanz`* ist.

Short name (Kurzname): lb

Kompatibilität: Eine HAProxy-Ebene ist mit den folgenden Ebenen kompatibel: Benutzerdefiniert, DB-Master und Memcached.

Offene Ports: HAProxy ermöglicht den öffentlichen Zugriff auf die Ports 22 (SSH), 80 (HTTP) und 443 (HTTPS).

Autoassign Elastic IP addresses (Elastic IP-Adressen automatisch zuweisen): Standardmäßig aktiviert

Default EBS volume (Standard-EBS-Volume): Nein

Standard-Sicherheitsgruppe: AWS- OpsWorks -LB-Server

Konfiguration: Um eine HAProxy-Schicht zu konfigurieren, müssen Sie Folgendes angeben:

- Zustandsprüfungs-URI (standardmäßig: http://*DNSName*/).
- Statistik-URI (standardmäßig: http://*DNSName*/haproxy?stats).
- Passwort für Statistik (optional).
- Methode für Zustandsprüfung (optional). Standardmäßig verwendet HAProxy die HTTP OPTIONS-Methode. Sie können auch GET oder HEAD angeben.
- Statistiken aktivieren (optional)
- Ports. Standardmäßig konfiguriert AWS OpsWorks Stacks HAProxy so, dass es sowohl HTTP- als auch HTTPS-Verkehr verarbeitet. Sie können HAProxy so konfigurieren, dass nur einer der beiden verarbeitet wird, indem Sie die Vorlage für Chef-Konfigurationen, `haproxy.cfg.erb`, überschreiben.

Setup recipes (Einrichtungsrezepte):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- vermeiden
- ebs
- opsworks_ganglia::client
- haproxy

Configure recipes (Konfigurationsrezepte):

- opsworks_ganglia::configure-client
- ssh_users

- agent_version
- haproxy::configure

Deploy recipes (Bereitstellungsrezepte):

- deploy::default
- haproxy::configure

Shutdown recipes (Shutdown-Rezepte):

- opsworks_shutdown::default
- haproxy::stop

Installation (Installation):

- AWS OpsWorks Stacks verwendet das Paketinstallationsprogramm der Instanz, um HAProxy an seinen Standardspeicherorten zu installieren.
- Sie müssen Syslog so einrichten, dass die Protokolldateien an einen bestimmten Speicherort geleitet werden. Weitere Informationen finden Sie unter HAProxy.

HAProxy Stacks AWS OpsWorks , Ebene

> **Note**
>
> Dieser Layer steht nur für Chef 11 und niedrigere Linux-basierte Stacks zur Verfügung.

Die AWS OpsWorks Stacks HAProxy-Schicht ist eine AWS OpsWorks Stacks-Schicht, die einen Blueprint für Instances bereitstellt, die einen HAProxy-Server hosten — ein zuverlässiger TCP/HTTP-Lastenausgleich mit hoher Leistung. Eine kleine Instance ist normalerweise ausreichend für die Verarbeitung des gesamten Datenverkehrs des Anwendungsservers.

> **Note**
>
> Stacks sind auf eine einzelne Region begrenzt. Um Ihre Anwendung über mehrere Regionen zu verteilen, müssen Sie für jede Region einen separaten Stack erstellen.

Um eine HAProxy-Schicht zu erstellen

1. Klicken Sie im Navigationsbereich auf Layers (Layers).
2. Klicken Sie auf der Seite Layers (Layers) auf Add a Layer (Einen Layer hinzufügen) oder auf + Layer (+ Layer). Wählen Sie für Layer type (Layer-Typ) die Option HAProxy (HAProxy) aus.

Der Layer verfügt über die folgenden Konfigurationseinstellungen, die alle optional sind.

HAProxy statistics (HAProxy-Statistiken)

Unabhängig davon, ob der Layer Statistiken sammelt oder anzeigt. Der Standardwert ist Yes.

Statistics URL (URL für Statistiken)

Der URL-Pfad der Statistikseite. Die vollständige URL lautet http://*dnsName StatisticsPath*, *wobei dnsName der DNS-Name* der zugehörigen Instanz ist. Der Standardwert ist /haproxy? *StatisticsPath* stats, was etwa entspricht: http://ec2-54-245-151-7.us-west-2.compute.amazonaws.com/haproxy?stats.

Statistics user name (Benutzername für Statistiken)

Der Benutzername der Statistikseite, den Sie angeben müssen, um die Statistikseite aufrufen zu können. Der Standardwert ist „opsworks“.

Statistics password (Passwort für Statistiken)

Ein Passwort für die Statistikseite, das Sie eingeben müssen, um die Statistikseite zu sehen. Der Standardwert ist eine zufällig erstellte Zeichenfolge.

Health check URL (URL für Zustandsprüfung)

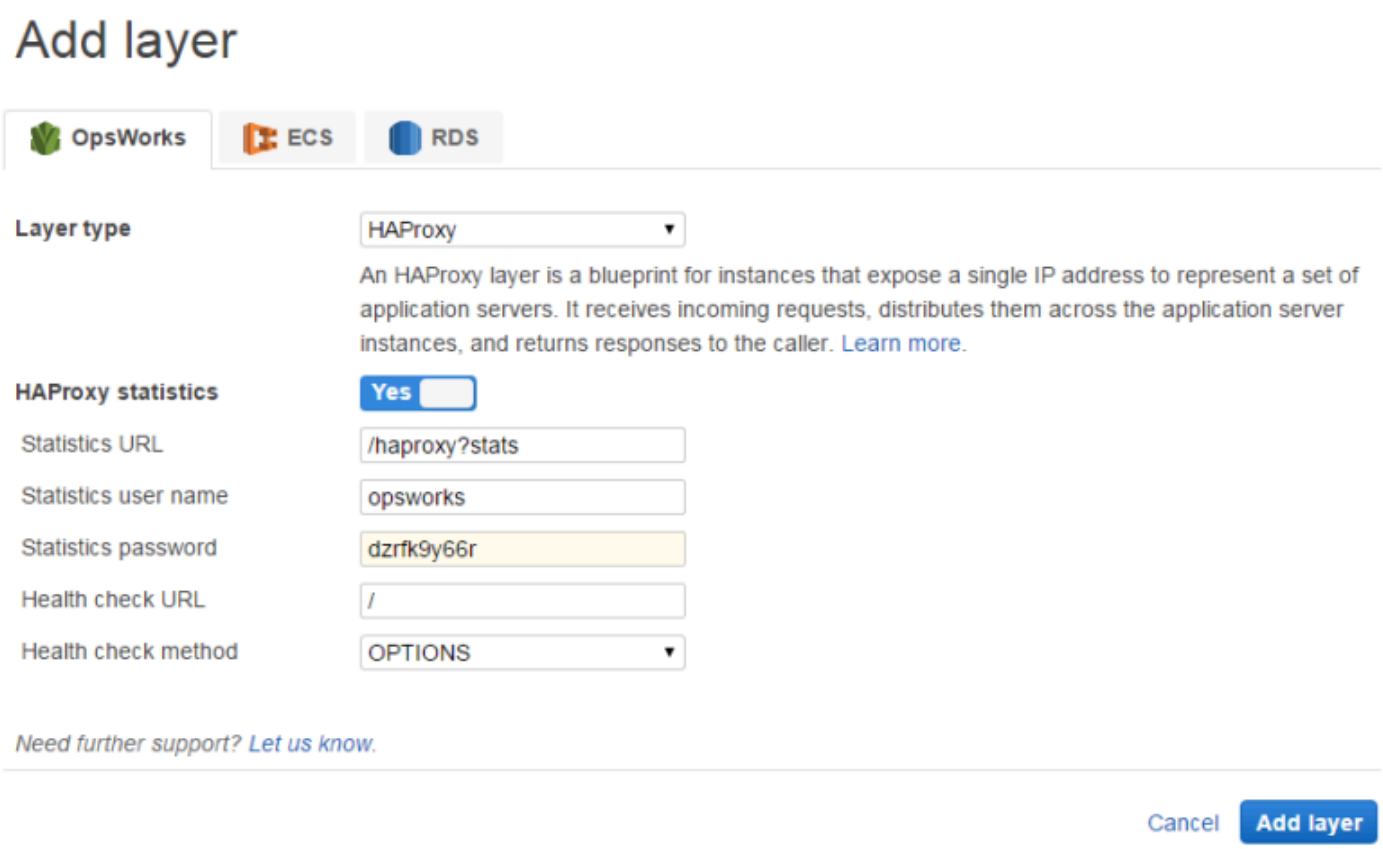
Das Zustandsprüfungs-URL-Suffix. HAProxy verwendet diese URL, um in regelmäßigen Abständen eine HTTP-Methode auf den einzelnen Anwendungsserver-Instances aufzurufen, um zu ermitteln, ob die Instance funktioniert. Wenn die Zustandsprüfung fehlschlägt, stoppt HAProxy das Routing von Datenverkehr an die Instance, bis sie neu gestartet ist, entweder manuell oder durch Auto Healing. Der Standardwert für das URL-Suffix lautet "/", was der Homepage der Server-Instance entspricht: http://*DNSName*/.

Health check method (Methode für Zustandsprüfung)

Eine HTTP-Methode, die in der Regel überprüft, ob Instances funktionieren. Der Standardwert ist OPTIONS und Sie können auch GET oder HEAD angeben. Weitere Informationen finden Sie unter httpchk.

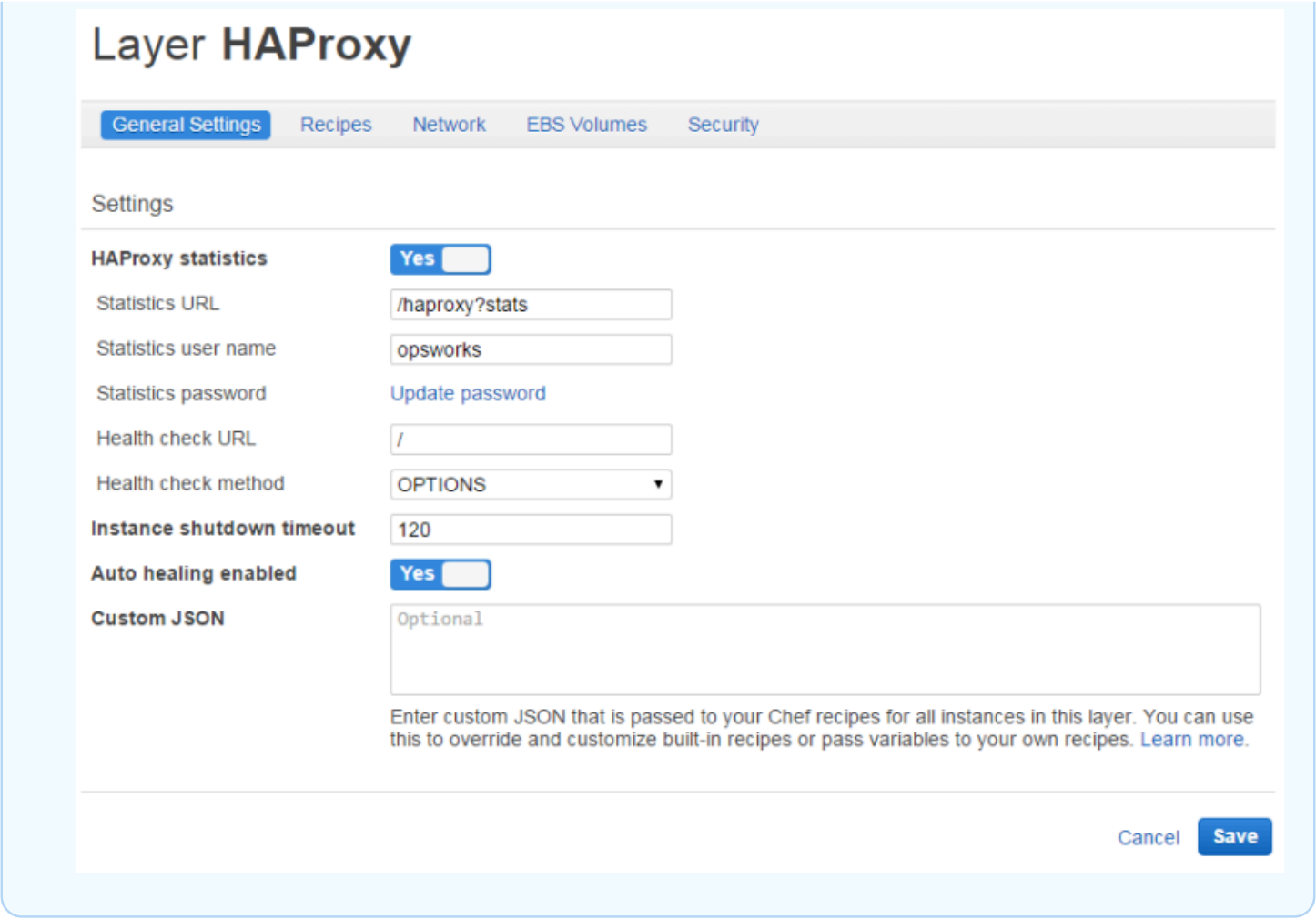
Benutzerdefinierte Sicherheitsgruppen

Diese Einstellung wird angezeigt, wenn Sie Ihren Layern nicht automatisch eine integrierte AWS OpsWorks Stacks-Sicherheitsgruppe zuordnen möchten. Sie müssen die mit der Ebene zu verknüpfende Sicherheitsgruppe angeben. Stellen Sie sicher, dass die Gruppe über die richtigen Einstellungen verfügt, um Datenverkehr zwischen den Layern zu erlauben. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.



Note

Notieren Sie sich das Passwort für die spätere Verwendung. In AWS OpsWorks Stacks können Sie das Passwort nicht anzeigen, nachdem Sie die Ebene erstellt haben. Sie können das Passwort jedoch aktualisieren, indem Sie auf die Seite Edit (Bearbeiten) des Layers wechseln und auf Update password (Passwort aktualisieren) auf der Registerkarte General Settings (Allgemeine Einstellungen) klicken.



So funktioniert die HAProxy-Ebene

HAProxy übernimmt standardmäßig folgende Aktionen:

- Empfängt Anfragen auf den HTTP- und HTTPS-Ports.

  Sie können HAProxy so konfigurieren, dass nur auf den HTTP- oder HTTPS-Port reagiert wird, indem Sie die Chef-Konfigurationsvorlage `haproxy.cfg.erb` überschreiben.
- Leitet den eingehenden Datenverkehr an Instances, die zu einem Anwendungsserver-Layer gehören.

  Standardmäßig konfiguriert AWS OpsWorks Stacks HAProxy so, dass der Datenverkehr an Instanzen verteilt wird, die Mitglieder einer beliebigen Anwendungsserverschicht sind. Sie könnten beispielsweise einen Stack mit den Ebenen Rails App Server und PHP App Server haben, und ein HAProxy-Master verteilt den Datenverkehr auf die Instanzen in beiden Schichten. Sie können

die Standard-Routing-Einstellungen konfigurieren, indem Sie eine benutzerdefiniertes Rezept verwenden.

- Verteilt den Datenverkehr auf mehrere Availability Zones.

  Wenn eine Availability Zone ausfällt, leitet der Load Balancer den eingehenden Datenverkehr an Instances in anderen Zonen, sodass Ihre Anwendung weiterhin ohne Unterbrechung ausgeführt wird. Aus diesem Grund ist es ein empfohlenes Verfahren, Ihre Anwendungsserver über mehrere Availability Zones hinweg zu verteilen.

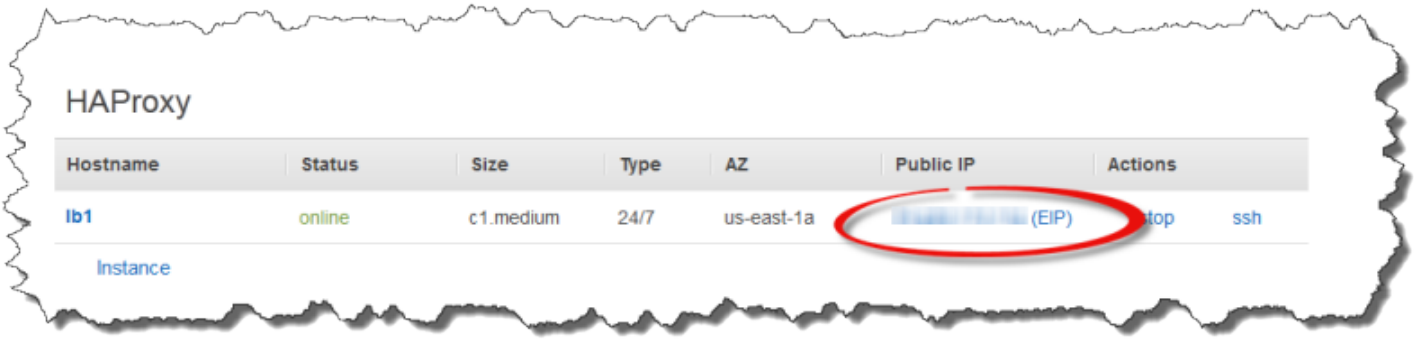
- Führt in regelmäßigen Abständen die angegebene Methode für Zustandsprüfungen auf jeder Anwendungsserver-Instance aus, um ihren Zustand zu bewerten.

  Wenn die Methode nicht innerhalb eines bestimmten Timeout-Zeitraums zurückkehrt, wird davon ausgegangen, dass die Instanz ausgefallen ist, und HAProxy stoppt das Weiterleiten von Anfragen an die Instanz. AWS OpsWorks Stacks bietet auch eine Möglichkeit, ausgefallene Instances automatisch zu ersetzen. Weitere Informationen finden Sie unter Verwenden von Auto Healing. Sie können die Methode der Zustandsprüfung beim Erstellen des Layers ändern.

- Sammelt Statistiken und zeigt sie optional auf einer Webseite.

> **Important**
>
> Damit die Zustandsprüfung einwandfrei mit der standardmäßig verwendeten OPTIONS-Methode funktioniert, muss Ihre Anwendung einen 2xx- oder 3xx-Statuscode zurückgeben.

Wenn Sie eine Instance zu einer HAProxy-Ebene hinzufügen, weist AWS OpsWorks Stacks ihr standardmäßig eine Elastic IP-Adresse zu, um die weltweit öffentliche Anwendung darzustellen. Da die Elastic IP-Adresse der HAProxy-Instanz die einzige öffentlich zugängliche URL der Anwendung ist, müssen Sie keine öffentlichen Domain-Namen für die zugrunde liegenden Anwendungsserver-Instanzen erstellen und verwalten. Sie erhalten die Adresse, indem Sie die Seite Instances (Instances) aufrufen und die öffentliche IP-Adresse überprüfen, wie in der folgenden Abbildung gezeigt. Eine Adresse, die von (EIP) gefolgt wird, ist eine Elastic IP-Adresse. Weitere Informationen zu Elastic IP-Adressen finden Sie unter Elastic IP Addresses (EIP) (Elastic IP-Adressen (EIP)).



Wenn Sie eine HAProxy-Instance beenden, behält AWS OpsWorks Stacks die Elastic IP-Adresse bei und weist sie der Instance neu zu, wenn Sie sie neu starten. Wenn Sie eine HAProxy-Instance löschen, löscht AWS OpsWorks Stacks standardmäßig die IP-Adresse der Instance. Um die Adresse zu bewahren, löschen Sie die Option zum Delete instance's Elastic IP (Löschen der Elastic IP der Instance), wie in der folgenden Abbildung dargestellt.

Diese Option wirkt sich auf das aus, was passiert, wenn Sie zu dem Layer eine neue Instance hinzufügen, um eine gelöschte Instance zu ersetzen:

- Wenn Sie die Elastic IP-Adresse der gelöschten Instance beibehalten haben, weist AWS OpsWorks Stacks die Adresse der neuen Instance zu.
- Andernfalls weist AWS OpsWorks Stacks der Instance eine neue Elastic IP-Adresse zu und Sie müssen Ihre DNS-Registrar-Einstellungen aktualisieren, damit sie der neuen Adresse zugeordnet werden.

Wenn Anwendungsserver-Instances online gehen oder offline gehen — entweder manuell oder als Folge der automatischen Skalierung oder auto Heilung — muss die Load Balancer-Konfiguration

aktualisiert werden, um den Datenverkehr an die aktuellen Online-Instanzen weiterzuleiten. Diese Aufgabe wird automatisch von den integrierten Rezepten des Layers durchgeführt:

- Wenn neue Instances online gehen, löst AWS OpsWorks Stacks ein Configure-Lifecycle-Ereignis aus. Die integrierten Configure-Rezepte der HAProxy-Schicht aktualisieren die Load Balancer-Konfiguration, sodass Anfragen auch an alle neuen Anwendungsserver-Instanzen verteilt werden.
- Wenn Instanzen offline gehen oder eine Instance eine Integritätsprüfung nicht besteht, löst AWS OpsWorks Stacks auch ein Configure-Lifecycle-Ereignis aus. Die HAProxy-Konfigurationsrezepte aktualisieren die Load Balancer-Konfiguration, um den Datenverkehr nur auf die verbleibenden Online-Instances zu leiten.

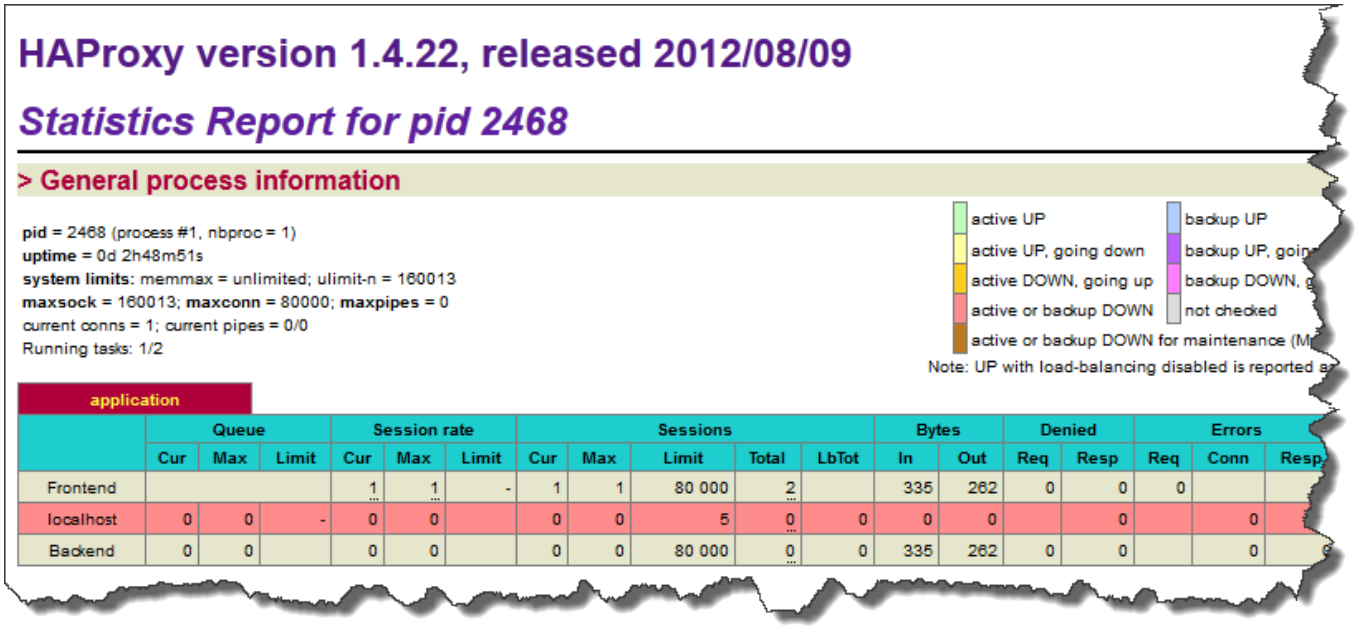
Schließlich können Sie auch eine benutzerdefinierte Domain mit der HAProxy-Ebene verwenden. Weitere Informationen finden Sie unter Verwenden von benutzerdefinierten Domänen.

Die Statistikseite

Wenn Sie die Statistikseite aktiviert haben, zeigt HAProxy eine Seite mit einer Vielzahl von Metriken unter der angegebenen URL an.

Um HAProxy-Statistiken anzuzeigen

1. Rufen Sie den öffentlichen DNS-Namen der HAProxy-Instanz von der Detailseite der Instanz ab und kopieren Sie ihn.
2. Klicken Sie auf der Seite Layers auf HAProxy, um die Detailseite des Layers zu öffnen.
3. Rufen Sie die Statistik-URL aus den Layer-Details ab und hängen Sie sie an den öffentlichen DNS-Namen an. Zum Beispiel: **`http://ec2-54-245-102-172.us-west-2.compute.amazonaws.com/haproxy?stats`**.
4. Fügen Sie die URL aus dem vorherigen Schritt in Ihren Browser ein und verwenden Sie den Benutzernamen und das Passwort, die Sie angegeben haben, als Sie den Layer erstellt haben, um die Statistikseite zu öffnen.



HAProxy version 1.4.22, released 2012/08/09

Statistics Report for pid 2468

> General process information

pid = 2468 (process #1, nbproc = 1)
uptime = 0d 2h48m51s
system limits: memmax = unlimited; ulimit-n = 160013
maxsock = 160013; maxconn = 80000; maxpipes = 0
current conns = 1; current pipes = 0/0
Running tasks: 1/2

active UP | backup UP
active UP, going down | backup UP, goin
active DOWN, going up | backup DOWN, g
active or backup DOWN | not checked
active or backup DOWN for maintenance (M
Note: UP with load-balancing disabled is reported a

application

	Queue			Session rate			Sessions					Bytes		Denied		Errors		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp
Frontend				1	1	-	1	1	80 000	2		335	262	0	0	0		
localhost	0	0	-	0	0		0	0	5	0	0	0	0		0		0	
Backend	0	0		0	0		0	0	80 000	0	0	335	262	0	0		0	

MySQL-Ebenenreferenz

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Die MySQL-Schicht unterstützt MySQL, ein weit verbreitetes relationales Datenbankverwaltungssystem. AWS OpsWorks Stacks installiert die neueste verfügbare Version, die vom Betriebssystem abhängt. Wenn Sie eine MySQL-Instance hinzufügen, werden den Anwendungsserver-Layern die benötigten Zugriffsinformationen bereitgestellt. Sie müssen benutzerdefinierte Chef-Rezepte schreiben, um Master-Master- oder Master-Slave-Konfigurationen einzurichten.

Short name (Kurzname): db-master

Kompatibilität: Eine MySQL-Schicht ist mit den folgenden Ebenen kompatibel: custom, lb, memcached, monitoring-master, nodejs-app, php-App, rails-app und web.

Offene Ports: Eine MySQL-Schicht ermöglicht den öffentlichen Zugriff auf Port 22 (SSH) und alle Ports von den Webservern, benutzerdefinierten Servern und den Anwendungsservern Rails, PHP und Node.js des Stacks.

Autoassign Elastic IP addresses (Elastic IP-Adressen automatisch zuweisen): Standardmäßig deaktiviert

Default EBS volume (Standard-EBS-Volume): Ja, bei `/vol/mysql`

Standard-Sicherheitsgruppe: AWS- OpsWorks -DB-Master-Server

Konfiguration: Um eine MySQL-Schicht zu konfigurieren, müssen Sie Folgendes angeben:

- Stammbenutzerpasswort
- MySQL-Engine

Setup recipes (Einrichtungsrezepte):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- vermeiden
- ebs
- opsworks_ganglia::client
- mysql::server
- vermeiden
- deploy::mysql

Configure recipes (Konfigurationsrezepte):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- deploy::mysql

Deploy recipes (Bereitstellungsrezepte):

- deploy::default
- deploy::mysql

Shutdown recipes (Shutdown-Rezepte):

- opsworks_shutdown::default
- mysql::stop

Installation (Installation):

- AWS OpsWorks Stacks verwendet das Paketinstallationsprogramm der Instanz, um MySQL und seine Protokolldateien an ihren Standardspeicherorten zu installieren. Weitere Informationen finden Sie in der MySQL-Dokumentation.

OpsWorks MySQL-Schicht

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Dieser Layer steht nur für Chef 11 und niedrigere Linux-basierte Stacks zur Verfügung.

Eine OpsWorks MySQL-Ebene bietet einen Blueprint für Amazon EC2 EC2-Instances, die als MySQL-Datenbankmaster fungieren. Mithilfe eines integrierten Rezepts erstellen Sie eine Datenbank für jede Anwendung, die für einen Anwendungsserver-Layer bereitgestellt wurde. Wenn Sie beispielsweise eine PHP-Anwendung "MeineApp" erstellen, erstellt die Vorlage eine "MeineApp"-Datenbank.

Die MySQL-Schicht hat die folgenden Konfigurationseinstellungen.

MySQL root user password (Passwort für MySQL-Root-Benutzer)

Passwort des Root-Benutzers (erforderlich)

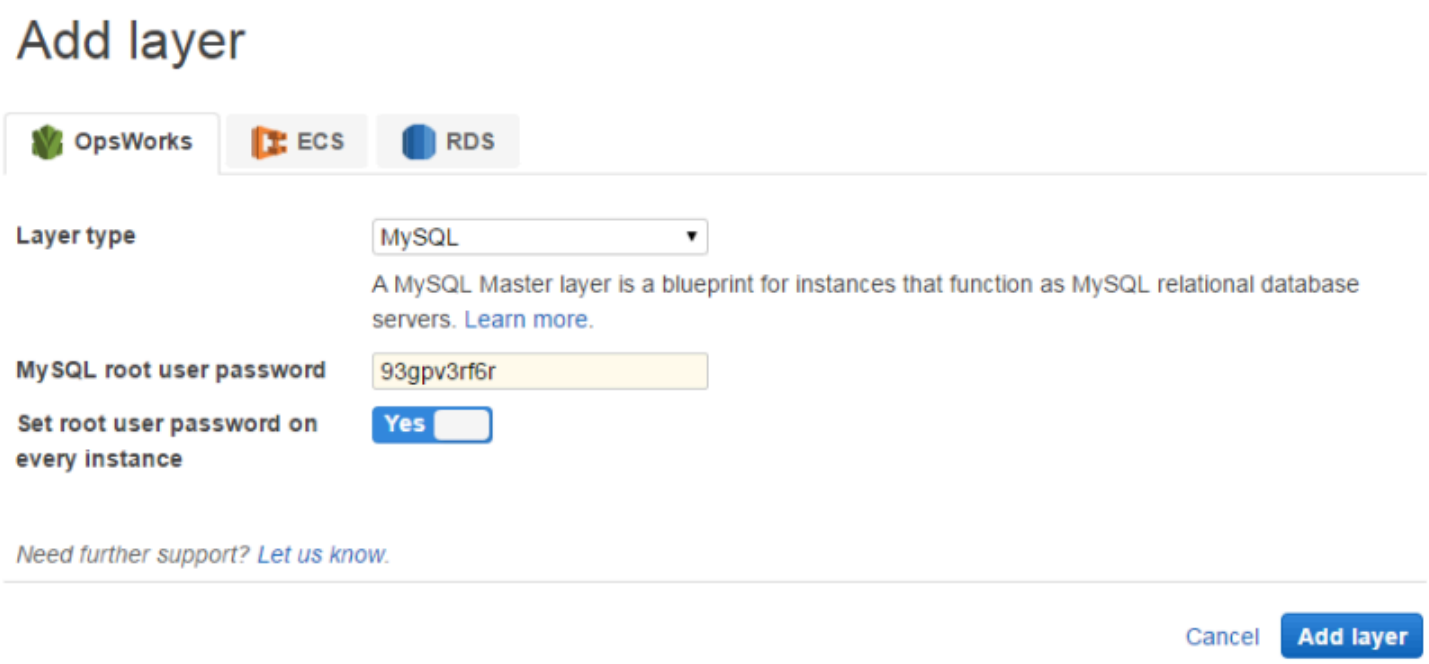
Set root user password on every instance (Root-Benutzer-Passwort für jede Instance setzen)

Legt fest, ob das Passwort des Root-Benutzers in den Stack-Konfigurations- und Bereitstellungsattributen enthalten ist, die für jede Instance des Stacks installiert werden (optional). Die Standardeinstellung ist Yes (Ja).

Wenn Sie diesen Wert auf Nein setzen, gibt AWS OpsWorks Stacks das Root-Passwort nur an Anwendungsserver-Instanzen weiter.

Benutzerdefinierte Sicherheitsgruppen

Eine benutzerdefinierte Sicherheitsgruppe, die dem Layer zugeordnet wird (optional). Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.



Sie können dem Layer eine oder mehrere Instances hinzufügen. Dabei repräsentiert jeder Layer eine eigene MySQL-Master-Datenbank. Fügen Sie dann einer App eine Instance hinzu, um die erforderlichen Verbindungsinformationen in den Anwendungsservern der App zu speichern. Die Anwendung kann nun die Verbindungsinformationen verwenden, um eine Verbindung zum Datenbankserver der Instance herzustellen.

Referenz für Anwendungsserver-Layer

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks unterstützt mehrere verschiedene Anwendungs- und statische Webseitenserver.

Themen

- AWS Flow (Ruby) Layer-Referenz

- Referenz zur Java-App-Serverschicht
- Referenz zur App-Serverschicht von Node.js
- Referenz zur PHP-App-Serverschicht
- Referenz zur Rails-App-Serverschicht
- Referenz auf statische Webserverebene

AWS Flow (Ruby) Layer-Referenz

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Eine AWS Flow (Ruby) -Ebene bietet einen Blueprint für Instances, die Amazon Simple Workflow Service-Aktivitäten und Workflow-Worker hosten.

Kurzname: aws-flow-ruby

Kompatibilität: Eine AWS Flow (Ruby) -Schicht ist mit PHP App Server, MySQL, Memcached, Ganglia und benutzerdefinierten Ebenen kompatibel.

Open ports (Offene Ports): Keine

IAM-Rolle: aws-opsworks-ec 2- role-with-swf ist die standardmäßige AWS Flow (Ruby) -Rolle, die AWS OpsWorks Stacks auf Anfrage für Sie erstellt.

Autoassign Elastic IP addresses (Elastic IP-Adressen automatisch zuweisen): Standardmäßig deaktiviert

Default EBS Volume (Standard EBS-Volume) Nein

Standard-Sicherheitsgruppe: AWS- OpsWorks -AWS-Flow-Ruby-Server

Setup recipes (Einrichtungsrezepte):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- vermeiden
- ebs
- opsworks_ganglia::client
- opsworks_aws_flow_ruby::setup

Configure recipes (Konfigurationsrezepte):

- opsworks_ganglia::configure-client
- ssh_users
- mysql::client
- agent_version
- opsworks_aws_flow_ruby::configure

Deploy recipes (Bereitstellungsrezepte):

- deploy::default
- bereitstellen:: aws-flow-ruby

Undeploy recipes (Bereitstellung von Rezepten aufheben):

- bereitstellen:: aws-flow-ruby-undeploy

Shutdown recipes (Shutdown-Rezepte):

- opsworks_shutdown::default

Referenz zur Java-App-Serverschicht

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Ein Java App Server-Layer unterstützt einen Apache Tomcat 7.0-Anwendungsserver.

Short name (Kurzname): java-app

Kompatibilität: Eine Java App Server-Schicht ist mit den folgenden Ebenen kompatibel: custom, db-master und memcached.

Offene Ports: Eine Java App Server-Schicht ermöglicht den öffentlichen Zugriff auf die Ports 22 (SSH), 80 (HTTP), 443 (HTTPS) und alle Ports von Load Balancers.

Autoassign Elastic IP addresses (Elastic IP-Adressen automatisch zuweisen): Standardmäßig deaktiviert

Default EBS Volume (Standard EBS-Volume) Nein

Standard-Sicherheitsgruppe: AWS- OpsWorks -Java-App-Server

Setup recipes (Einrichtungsrezepte):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users

- mysql::client
- vermeiden
- ebs
- opsworks_ganglia::client
- opsworks_java::setup

Configure recipes (Konfigurationsrezepte):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- opsworks_java::configure

Deploy recipes (Bereitstellungsrezepte):

- deploy::default
- deploy::java

Undeploy recipes (Bereitstellung von Rezepten aufheben):

- deploy::java-undeploy

Shutdown recipes (Shutdown-Rezepte):

- opsworks_shutdown::default
- deploy::java-stop

Installation (Installation):

- Tomcat wird in `/usr/share/tomcat7` installiert.
- Weitere Informationen dazu, wie Protokolldateien erstellt werden, finden Sie unter Logging in Tomcat.

Referenz zur App-Serverschicht von Node.js

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Ein App-Server-Layer von Node.js unterstützt einen Anwendungsserver vom Typ Node.js. Dabei handelt es sich um eine Plattform für die Implementierung hoch skalierbarer Netzwerkanwendungsserver. Programme werden mithilfe ereignisgesteuerter asynchroner I/O geschrieben JavaScript, um den Overhead zu minimieren und die Skalierbarkeit zu maximieren.

Short name (Kurzname): nodejs-app

Kompatibilität: Eine App Server-Ebene von Node.js ist mit den folgenden Ebenen kompatibel: Benutzerdefiniert, DB-Master, Memcached und Monitoring-Master.

Offene Ports: Eine Node.js App Server-Ebene ermöglicht den öffentlichen Zugriff auf die Ports 22 (SSH), 80 (HTTP), 443 (HTTPS) und alle Ports von Load Balancers.

Autoassign Elastic IP addresses (Elastic IP-Adressen automatisch zuweisen): Standardmäßig deaktiviert

Default EBS volume (Standard-EBS-Volume): Nein

Standard-Sicherheitsgruppe: AWS- OpsWorks -NodeJS-App-Server

Setup recipes (Einrichtungsrezepte):

- opsworks_initial_setup
- ssh_host_keys

- ssh_users
- mysql::client
- vermeiden
- ebs
- opsworks_ganglia::client
- opsworks_nodejs
- opsworks_nodejs::npm

Configure recipes (Konfigurationsrezepte)

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- opsworks_nodejs::configure

Deploy recipes (Bereitstellungsrezepte):

- deploy::default
- opsworks_nodejs
- opsworks_nodejs::npm
- deploy::nodejs

Undeploy recipes (Bereitstellung von Rezepten aufheben):

- deploy::nodejs-undeploy

Shutdown recipes (Shutdown-Rezepte):

- opsworks_shutdown::default
- deploy::nodejs-stop

Installation (Installation):

- Node.js wird in `/usr/local/bin/node` installiert.

- Weitere Informationen zur Erstellung von Protokolldateien finden Sie unter How to log in node.js auf der Nodejitsu-Website.

Node.js application configuration (Node.js-Anwendungskonfiguration):

- Die von Node.js ausgeführte Hauptdatei muss `server.js` benannt werden und sich im Stammverzeichnis der bereitgestellten Anwendung befinden.
- Die Node.js-Anwendung muss so festgelegt sein, dass Port 80 (oder Port 443, falls zutreffend) verwendet wird.

> **Note**
>
> Node.js-Apps, die Express ausführen, verwenden normalerweise den folgenden Code, um den Überwachungsport festzulegen, wobei `process.env.PORT` den Standardport darstellt und in 80 aufgelöst wird:
>
> ```
> app.set('port', process.env.PORT || 3000);
> ```
>
> Bei AWS OpsWorks Stacks müssen Sie Port 80 explizit wie folgt angeben:
>
> ```
> app.set('port', 80);
> ```

Referenz zur PHP-App-Serverschicht

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Die PHP App Server-Ebene unterstützt einen PHP-Anwendungsserver unter Verwendung von Apache2 mit mod_php.

Short name (Kurzname): php-app

Kompatibilität: Ein PHP-App-Server-Layer ist mit den folgenden Ebenen kompatibel: custom, db-Master, memcached, Monitoring-Master und Rails-App.

Offene Ports: Eine PHP-App-Server-Schicht ermöglicht den öffentlichen Zugriff auf die Ports 22 (SSH), 80 (HTTP), 443 (HTTPS) und alle Ports von Load Balancers.

Autoassign Elastic IP addresses (Elastic IP-Adressen automatisch zuweisen): Standardmäßig deaktiviert

Default EBS volume (Standard-EBS-Volume): Nein

Standard-Sicherheitsgruppe: AWS- OpsWorks -PHP-App-Server

Setup recipes (Einrichtungsrezepte):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- vermeiden
- ebs
- opsworks_ganglia::client
- mysql::client
- vermeiden
- mod_php5_apache2

Configure recipes (Konfigurationsrezepte):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- mod_php5_apache2::php
- php::configure

Deploy recipes (Bereitstellungsrezepte):

- deploy::default
- deploy::php

Undeploy recipes (Bereitstellung von Rezepten aufheben):

- deploy::php-undeploy

Shutdown recipes (Shutdown-Rezepte):

- opsworks_shutdown::default
- apache2::stop

Installation (Installation):

- AWS OpsWorks Stacks verwendet das Paketinstallationsprogramm der Instanz, um Apache2, mod_php und die zugehörigen Protokolldateien an ihren Standardspeicherorten zu installieren. Weitere Informationen zur Installation finden Sie unter Apache. Weitere Informationen zur Protokollierung finden Sie unter Log Files.

Referenz zur Rails-App-Serverschicht

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Die Rails App Server-Schicht unterstützt einen Ruby on Rails-Anwendungsserver.

Short name (Kurzname): rails-app

Kompatibilität: Eine Rails App Server-Schicht ist mit den folgenden Schichten kompatibel: Benutzerdefiniert, DB-Master, Memcached, Monitoring-Master, PHP-App.

Ports: Eine Rails App Server-Schicht ermöglicht den öffentlichen Zugriff auf die Ports 22 (SSH), 80 (HTTP), 443 (HTTPS) und alle Ports von Load Balancers.

Autoassign Elastic IP addresses (Elastic IP-Adressen automatisch zuweisen): Standardmäßig deaktiviert

Default EBS volume (Standard-EBS-Volume): Nein

Standard-Sicherheitsgruppe: AWS- OpsWorks -Rails-App-Server

Konfiguration: Um einen Rails App Server-Layer zu konfigurieren, müssen Sie Folgendes angeben:

- Ruby-Version
- Rails-Stack
- Rubygems-Version
- Ob Bundler installiert und verwaltet werden soll
- Die Bundler-Version

Setup recipes (Einrichtungsrezepte):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users

- mysql::client
- vermeiden
- ebs
- opsworks_ganglia::client
- apache2 apache2::mod_deflate
- passenger_apache2
- passenger_apache2::mod_rails
- passenger_apache2::rails

Configure recipes (Konfigurationsrezepte):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- rails::configure

Deploy recipes (Bereitstellungsrezepte)

- deploy::default
- deploy::rails

Undeploy recipes (Bereitstellung von Rezepten aufheben):

- deploy::rails-undeploy

Shutdown recipes (Shutdown-Rezepte):

- opsworks_shutdown::default
- apache2::stop

Installation (Installation):

- AWS OpsWorks Stacks verwendet das Paketinstallationsprogramm der Instanz, um Apache2 mit mod_passenger, mod_rails und den zugehörigen Protokolldateien an ihren Standardspeicherorten

zu installieren. Weitere Informationen über die Installation finden Sie unter Phusion Passenger. Weitere Informationen zur Protokollierung finden Sie unter Log Files.

Referenz auf statische Webserverebene

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Die statische Webserver-Ebene dient statischen HTML-Seiten, die clientseitigen Code enthalten können, wie z. JavaScript Er basiert auf Nginx, einem Open-Source-HTTP-, Reverse-Proxy- und Mail-Proxy-Server.

Short name (Kurzname): web

Kompatibilität: Eine statische Webserver-Ebene ist mit den folgenden Ebenen kompatibel: Benutzerdefiniert, DB-Master, Memcached.

Offene Ports: Eine statische Webserver-Schicht ermöglicht den öffentlichen Zugriff auf die Ports 22 (SSH), 80 (HTTP), 443 (HTTPS) und alle Ports von Load Balancers.

Autoassign Elastic IP addresses (Elastic IP-Adressen automatisch zuweisen): Standardmäßig deaktiviert

Default EBS volume (Standard-EBS-Volume): Nein

Standard-Sicherheitsgruppe: AWS- OpsWorks -Web-Server

Setup recipes (Einrichtungsrezepte):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- vermeiden
- ebs
- opsworks_ganglia::client
- nginx

Configure recipes (Konfigurationsrezepte):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version

Deploy recipes (Bereitstellungsrezepte):

- deploy::default
- deploy::web

Undeploy recipes (Bereitstellung von Rezepten aufheben):

- deploy::web-undeploy

Shutdown recipes (Shutdown-Rezepte):

- opsworks_shutdown::default
- nginx::stop

Installation (Installation):

- Nginx wird in `/usr/sbin/nginx` installiert.
- Nginx-Protokolldateien sind in `/var/log/nginx`.

Anwendungsserverebene

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebenen sind nur für Chef 11 und frühere Linux-basierte Stacks verfügbar.

AWS OpsWorks Stacks unterstützt mehrere verschiedene Anwendungsserver, wobei der Begriff „Anwendung“ statische Webseiten umfasst. Jeder Servertyp verfügt über eine separate AWS OpsWorks Stacks-Ebene mit integrierten Rezepten, die die Installation des Anwendungsservers und aller zugehörigen Pakete auf den einzelnen Instanzen der Ebene, die Bereitstellung von Apps usw. übernehmen. Die Java App Server-Schicht installiert beispielsweise mehrere Pakete — darunter Apache, Tomcat und OpenJDK — und stellt Java-Apps auf jeder Instanz der Ebene bereit.

Nachfolgend wird das grundsätzliche Verfahren zur Verwendung einer Anwendungsserverebene beschrieben:

1. Erstellen Sie eine der verfügbaren App Server-Ebenentypen.
2. Fügen Sie eine oder mehrere Instances der Ebene hinzu.
3. Erstellen Sie Anwendungen und stellen Sie sie den Instances bereit. Weitere Informationen finden Sie unter Apps.
4. (Optional) Wenn die Ebene über mehrere Instances verfügt, können Sie einen Load Balancer hinzufügen, der den eingehenden Datenverkehr an die Instances verteilt. Weitere Informationen finden Sie unter HAProxy Stacks AWS OpsWorks , Ebene.

Themen

- AWS-Flow-Schicht (Ruby)
- AWS OpsWorks Stacks-Schicht für Java App Server

- Node.js App Server AWS OpsWorks Stacks Layer
- PHP-App-Server: AWS OpsWorks Stacks-Layer
- Rails App Server: AWS OpsWorks Stacks-Ebene
- Statischer Webserver: AWS OpsWorks Stacks Layer

AWS-Flow-Schicht (Ruby)

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Eine AWS Flow (Ruby) -Ebene ist eine AWS OpsWorks Stacks-Ebene, die einen Blueprint für Instances bereitstellt, die Amazon SWF SWF-Aktivitäten und Workflow-Worker hosten. Die Worker werden mithilfe des AWS Flow Framework for Ruby implementiert, einem Programmier-Framework, das den Prozess der Implementierung einer verteilten asynchronen Anwendung vereinfacht und gleichzeitig alle Vorteile von Amazon SWF bietet. Es ist ideal für die Implementierung von Anwendungen in verschiedensten Szenarien, einschließlich Geschäftsprozessen, Media-Kodierung, Aufgaben mit langen Ausführungszeiten und Hintergrundverarbeitung.

Die AWS Flow (Ruby) -Schicht umfasst die folgenden Konfigurationseinstellungen.

RubyGems Version

Die Framework-Version des Gems.

Bundler-Version

Die Bundler-Version.

EC2-Instance-Profil

> Ein benutzerdefiniertes Amazon EC2 EC2-Instance-Profil, das von den Instances des Layers verwendet werden soll. Dieses Profil muss Anwendungen, die auf den Instances des Layers ausgeführt werden, Berechtigungen für den Zugriff auf Amazon SWF gewähren.

Wenn Ihr Konto kein geeignetes Profil hat, können Sie Neues Profil mit SWF-Zugriff auswählen, damit AWS OpsWorks Stacks das Profil aktualisiert, oder Sie können es selbst mithilfe der IAM-Konsole aktualisieren. Das aktualisierte Profil können Sie anschließend für alle nachfolgenden AWS Flow-Ebenen verwenden. Im Folgenden finden Sie eine kurze Beschreibung, wie Sie das Profil mithilfe der IAM-Konsole erstellen. Weitere Informationen finden Sie unter Identity and Access Management in Amazon Simple Workflow Service.

Erstellen eines Profils für AWS Flow (Ruby) -Instances

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Richtlinien und dann Richtlinie erstellen aus, um eine neue vom Kunden verwaltete Richtlinie zu erstellen.
3. Wählen Sie für Service die Option SWF aus.
4. Wählen Sie für Aktionen die Option Alle SWF-Aktionen (swf: *).
5. Geben Sie für Amazon Resource Name (ARN) den ARN ein, der angibt, auf welche Amazon SWF-Domänen die Worker zugreifen können. Wählen Sie aus**`All resources`**, ob Sie Zugriff auf alle Domänen gewähren möchten.
6. Wählen Sie Weiter aus.
7. Geben Sie optional ein Tag ein, um die Richtlinie zu identifizieren.
8. Wählen Sie Weiter aus.
9. Wenn Sie fertig sind, wählen Sie Richtlinie erstellen aus.
10. Wählen Sie im Navigationsbereich Rollen und dann Rolle erstellen aus.
11. Geben Sie den Rollennamen an und wählen Sie Next Step aus. Sie können den Namen nicht ändern, nachdem die Rolle erstellt wurde.
12. Wählen Sie AWS-Service und dann EC2.
13. Wählen Sie Weiter aus.
14. Wählen Sie aus der Liste der Berechtigungsrichtlinien die Richtlinie aus, die Sie zuvor erstellt haben.

15. Wählen Sie Weiter aus.
16. Geben Sie einen Rollennamen ein und klicken Sie auf Create Role (Rolle erstellen). Sie können den Namen nicht ändern, nachdem die Rolle erstellt wurde.
17. Geben Sie dieses Profil an, wenn Sie eine AWS Flow (Ruby) -Layer in AWS OpsWorks Stacks erstellen.

AWS OpsWorks Stacks-Schicht für Java App Server

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Die Java App Server-Schicht ist eine AWS OpsWorks Stacks-Schicht, die einen Entwurf für Instanzen bereitstellt, die als Java-Anwendungsserver fungieren. Diese Schicht basiert auf Apache Tomcat 7.0 und Open JDK 7. AWS OpsWorks Stacks installiert auch die Java-Connector-Bibliothek, die es Java-Apps ermöglicht, ein `DataSource` JDBC-Objekt zu verwenden, um eine Verbindung zu einem Back-End-Datenspeicher herzustellen.

Installation: Tomcat wird installiert in `/usr/share/tomcat7`.

Auf der Seite Add Layer (Ebene hinzufügen) stehen folgende Konfigurationsoptionen zur Verfügung:

Java-VM-Optionen

Mit dieser Einstellung können Sie die benutzerdefinierten Java-VM-Optionen definieren. Es existieren keine Standard-Optionen. Ein oft verwendeter Optionssatz lautet `-Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC`. Wenn Sie Java VM Options verwenden, stellen Sie sicher, dass Sie einen gültigen Satz von Optionen übergeben.

AWS OpsWorks Stacks validiert die Zeichenfolge nicht. Wenn Sie versuchen, eine ungültige Option anzugeben, startet der Tomcat-Server in der Regel nicht, sodass die Einrichtung fehlschlägt. In diesem Fall können Sie dem Chef-Setup-Protokoll der Instance weitere Details entnehmen. Weitere Informationen zur Anzeige und Deutung der Chef-Protokolle finden Sie unter Chef-Protokolle.

Benutzerdefinierte Sicherheitsgruppen

Diese Einstellung wird angezeigt, wenn Sie Ihren Layern nicht automatisch eine integrierte AWS OpsWorks Stacks-Sicherheitsgruppe zuordnen möchten. Sie müssen die mit der Ebene zu verknüpfende Sicherheitsgruppe angeben. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

Elastic Load Balancer

Sie können den Instances des Layers einen Elastic Load Balancing Load Balancer zuordnen. Weitere Informationen finden Sie unter Elastic Load Balancing Lastenausgleichsebene.

Mit einem benutzerdefinierten JSON-Objekt oder einer benutzerdefinierten Attributdatei können Sie andere Konfigurationseinstellungen angeben. Weitere Informationen finden Sie unter Benutzerdefinierte Konfiguration.

> **Important**
>
> Wenn Ihre Java-Anwendung SSL verwendet, empfehlen wir, SSLv3 zu deaktivieren, um die in CVE-2014-3566 beschriebenen Schwachstellen zu vermeiden. Weitere Informationen finden Sie unter Deaktivieren von SSLv3 für Apache-Server.

Themen

- Deaktivieren von SSLv3 für Apache-Server
- Benutzerdefinierte Konfiguration
- Bereitstellen von Java-Anwendungen

Deaktivieren von SSLv3 für Apache-Server

Zum Deaktivieren von SSLv3 müssen Sie in der Datei `ssl.conf` im Apache-Server die Einstellung `SSLProtocol` ändern. Dazu müssen Sie die `ssl.conf.erb` Vorlagendatei des integrierten

Apache2-Kochbuchs überschreiben, die in den Setup-Rezepten des Java App Server-Layers erstellt wird. `ssl.conf` Die Details hängen davon ab, welches Betriebssystem Sie für die Ebenen-Instances angeben. Die folgende Übersicht zeigt die erforderlichen Änderungen für Amazon Linux- und Ubuntu-Systeme. SSLv3 wird für Red Hat Enterprise Linux (RHEL)-Systeme automatisch deaktiviert. Weitere Informationen zum Überschreiben einer integrierten Vorlage finden Sie unter Verwenden von benutzerdefinierten Vorlagen.

Amazon Linux

Die Datei `ssl.conf.erb` für diese Betriebssysteme befindet sich im Verzeichnis `apache2` cookbook's `apache2/templates/default/mods`. Das folgende Beispiel zeigt den relevanten Teil der integrierten Datei.

```
...
#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>
```

Überschreiben Sie `ssl.conf.erb` und ändern Sie die `SSLProtocol`-Einstellung folgendermaßen.

```
...
#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

Die Datei `ssl.conf.erb` für dieses Betriebssystem befindet sich im Verzeichnis `apache2` cookbook's `apache2/templates/ubuntu-14.04/mods`. Das folgende Beispiel zeigt den relevanten Teil der integrierten Datei.

```
...
# The protocols to enable.
```

```
# Available values: all, SSLv3, TLSv1.2
# SSL v2 is no longer supported
SSLProtocol all
...
```

Ändern Sie diese Einstellung folgendermaßen.

```
...
# The protocols to enable.
# Available values: all, SSLv3, TLSv1.2
# SSL v2 is no longer supported
SSLProtocol all -SSLv3 -SSLv2
...
```

Benutzerdefinierte Konfiguration

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks stellt zusätzliche Konfigurationseinstellungen als integrierte Attribute zur Verfügung, die sich alle im Namespace befinden. `opsworks_java` Sie können eine benutzerdefinierte JSON- oder Attributdatei angeben, um die integrierten Attribute zu überschreiben und benutzerdefinierte Werte anzugeben. Die JVM- und Tomcat-Versionen werden beispielsweise von den integrierten `jvm_version`- und `java_app_server_version`-Attributen repräsentiert, die auf den Wert 7 gesetzt sind. Sie können eine oder beide mithilfe einer benutzerdefinierte JSON- oder Attributdatei auf den Wert 6 setzen. In dem folgenden Beispiel werden beide Attribute mithilfe einer benutzerdefinierten JSON-Datei auf den Wert 6 gesetzt:

```
{
  "opsworks_java": {
    "jvm_version": 6,
```

```
    "java_app_server_version" : 6
  }
}
```

Weitere Informationen finden Sie unter Nutzen eines benutzerdefinierten JSON-Objekts.

Ein weiteres Beispiel für eine benutzerdefinierte Konfiguration besteht in der Installation eines benutzerdefinierten JDK, um die Attribute `use_custom_pkg_location` und `custom_pkg_location_url_debian`, `custom_pkg_location_url_rhel` zu überschreiben.

> **Note**
>
> Wenn Sie die integrierten Rezeptbücher überschreiben, müssen Sie diese Komponenten selbst aktualisieren.

Weitere Informationen zu Attributen und wie diese überschrieben werden können, finden Sie unter Überschreiben der Attribute. Eine Liste der integrierten Attribute finden Sie unter opsworks_java-Attribute.

Bereitstellen von Java-Anwendungen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In den folgenden Themen wird beschrieben, wie Apps auf den Instanzen eines Java App Server-Layers bereitgestellt werden. In den Beispielen werden JSP-Anwendungen verwendet, aber Sie können zum Installieren von anderen Java-Anwendungsarten grundsätzlich in gleicher Weise vorgehen.

Sie können JSP-Seiten aus einem der unterstützten Repositorys bereitstellen. Wenn Sie WAR-Dateien bereitstellen möchten, beachten Sie, dass AWS OpsWorks Stacks automatisch WAR-Dateien extrahiert, die aus einem Amazon S3- oder HTTP-Archiv bereitgestellt werden, aber nicht

aus einem Git- oder Subversion-Repository. Wenn Sie Git oder Subversion für WAR-Dateien verwenden möchten, können Sie einen der folgenden Schritte ausführen:

- Speichern Sie das extrahierte Archiv im Repository.
- Speichern Sie die WAR-Datei im Repository und verwenden Sie zum Extrahieren des Archivs einen Chef-Bereitstellungs-Hook, wie im nachstehenden Beispiel beschrieben.

Sie können Chef-Bereitstellungs-Hooks verwenden, um vom Benutzer angegebene Ruby-Anwendungen in einer Instance in irgendeiner der vier Bereitstellungsstufen auszuführen. Die Phase wird durch den Anwendungsnamen bestimmt. Im folgenden Beispiel wird eine Ruby-Anwendung mit dem Namen `before_migrate.rb` beschrieben, die eine von einem Git- oder Subversion-Repository bereitgestellte WAR-Datei extrahiert. Der Name verknüpft die Anwendung mit dem Checkout-Bereitstellungs-Hook, sodass dieser zu Beginn der Bereitstellung nach erfolgter Prüfung des Codes, aber vor der Migration ausgeführt wird. Weitere Informationen zur Verwendung dieses Beispiels finden Sie unter Verwenden von Chef-Bereitstellungs-Hooks.

```
::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**
>
> Wenn Sie eine Aktualisierung für eine JSP-Anwendung bereitstellen, wird Tomcat die Aktualisierung möglicherweise nicht erkennen und stattdessen die vorhandene Anwendungsversion ausführen. Dies kann beispielsweise auftreten, wenn Sie die Anwendung als ZIP-Datei bereitstellen, die nur eine JSP-Seite enthält. Um sicherzustellen, dass Tomcat die zuletzt bereitgestellte Version ausführt, muss das Stammverzeichnis des Projekts das Verzeichnis "WEB-INF" mit einer Datei `web.xml` enthalten. Der Inhalt einer `web.xml`-Datei kann vielfältig sein. Nachstehendes ist jedoch ausreichend, um sicherzustellen, dass Tomcat die Aktualisierungen erkennt und die aktuell bereitgestellte Anwendungsversion ausführt. Sie müssen die Version nicht für jede Aktualisierung ändern. Tomcat erkennt die Aktualisierung auch dann, wenn sich die Version nicht geändert hat.

```
<context-param>
  <param-name>appVersion</param-name>
  <param-value>0.1</param-value>
</context-param>
```

Themen

- Bereitstellen einer JSP-Anwendung
- Bereitstellen einer JSP-Anwendung mit einer Backend-Datenbank

Bereitstellen einer JSP-Anwendung

Geben Sie zur Bereitstellung einer JSP-Anwendung den Namen und die Repository-Daten an. Außerdem können Sie optional Domänen und SSL-Einstellungen angeben. Weitere Informationen zum Erstellen einer Anwendung finden Sie unter Hinzufügen von Apps. Das folgende Verfahren zeigt, wie Sie eine einfache JSP-Seite aus einem öffentlichen Amazon S3 S3-Archiv erstellen und bereitstellen. Informationen zur Verwendung anderer Repository-Typen, einschließlich privater Amazon S3 S3-Archive, finden Sie unterAnwendungsquelle.

Das folgende Beispiel zeigt die JSP-Seite, auf der lediglich einige Systeminformationen aufgeführt sind.

```
<%@ 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**
>
> Im folgenden Verfahren wird davon ausgegangen, dass Sie bereits mit den Grundlagen zum Erstellen von Stacks, Hinzufügen von Instances zu Ebenen usw. vertraut sind. Wenn Sie AWS OpsWorks Stacks noch nicht kennen, sollten Sie zuerst nachlesenErste Schritte mit Chef 11 Linux-Stacks.

So stellen Sie eine JSP-Seite aus einem Amazon S3 S3-Archiv bereit

1. Erstellen Sie einen Stack mit einem Java App Server-Layer, fügen Sie dem Layer eine 24/7-Instance hinzu und starten Sie ihn.
2. Kopieren Sie den Code in eine Datei mit dem Namen `simplejsp.jsp`, speichern Sie die Datei in einem Verzeichnis mit dem Namen `simplejsp` und erstellen Sie ein `.zip`-Archiv des Verzeichnisses. Die Namen sind willkürlich. Sie können beliebige Datei- oder Verzeichnisnamen verwenden. Sie können auch andere Archivtypen verwenden, einschließlich gzip, bzip2, Tarball oder Java WAR. Beachten Sie, dass AWS OpsWorks Stacks keine unkomprimierten Tarballs unterstützt. Um mehrere JSP-Seiten bereitzustellen, fügen Sie diese zu demselben Archiv hinzu.
3. Laden Sie das Archiv in einen Amazon S3 S3-Bucket hoch und veröffentlichen Sie die Datei. Kopieren Sie die URL der Datei zur späteren Verwendung. Weitere Informationen zum Erstellen von Buckets und Hochladen von Dateien finden Sie unter Erste Schritte mit Amazon Simple Storage Service
4. Klicken Sie auf Hinzufügen einer Anwendung zum Stack und geben Sie folgende Einstellungen an:

   - Name (Name – `SimpleJSP`
   - App type – `Java`
   - Repository-Typ – `Http Archive`
   - Repository-URL — die Amazon S3 S3-URL Ihrer Archivdatei.

   Verwenden Sie die Standardwerte für die übrigen Einstellungen und klicken Sie anschließend auf Add App (App hinzufügen), um die Anwendung zu erstellen.
5. Stellen Sie die App auf der Java App Server-Instance bereit.

Sie können jetzt die URL der Anwendung aufrufen und die Anwendung anzeigen. Wenn Sie keine Domäne angegeben haben, können Sie eine URL erstellen, indem Sie entweder die öffentliche IP-Adresse und den öffentlichen DNS-Namen der Instance verwenden. Um die öffentliche IP-Adresse oder den öffentlichen DNS-Namen einer Instanz abzurufen, gehen Sie zur AWS OpsWorks Stacks-Konsole und klicken Sie auf der Seite Instances auf den Namen der Instanz, um deren Detailseite zu öffnen.

Der Rest der URL hängt vom Kurznamen der App ab. Dabei handelt es sich um einen Namen in Kleinbuchstaben, den AWS OpsWorks Stacks aus dem Namen der App generiert, den Sie bei der Erstellung der App angegeben haben. Der Kurzname von SimpleJSP, beispielsweise, lautet simplejsp. Sie können den Kurznamen einer Anwendung aus ihrer Detailseite entnehmen.

- Wenn der Kurzname `root` lautet, können Sie entweder `http://`*`public_DNS`*`/`*`appname`*`.jsp` oder `http://`*`public_IP`*`/`*`appname`*`.jsp` verwenden.
- Andernfalls können Sie entweder `http://`*`public_DNS`*`/`*`app_shortname`*`/`*`appname`*`.jsp` oder `http://`*`public_IP`*`/`*`app_shortname`*`/`*`appname`*`.jsp` verwenden.

Wenn Sie eine Domäne für die Anwendung angegeben haben, lautet die URL `http://`*`domain`*`/`*`appname`*`.jsp`.

Die URL sollte in etwa wie folgt aussehen: `http://192.0.2.0/simplejsp/simplejsp.jsp`.

Wenn Sie mehrere Anwendungen für dieselbe Instance bereitstellen möchten, dürfen Sie `root` nicht als Kurznamen verwenden. Dies kann URL-Konflikte verursachen, die verhindern, dass die Anwendung ordnungsgemäß funktioniert. Weisen Sie stattdessen den Anwendungen jeweils unterschiedliche Domänennamen zu.

Bereitstellen einer JSP-Anwendung mit einer Backend-Datenbank

JSP-Seiten können ein JDBC-`DataSource`-Objekt verwenden, um eine Verbindung mit einer Backend-Datenbank zu erstellen. Sie können eine Anwendung erstellen und bereitstellen, indem Sie das im vorherigen Abschnitt beschriebene Verfahren anwenden. Dabei ist ein zusätzlicher Schritt auszuführen, um die Verbindung einzurichten.

Auf der folgenden JSP-Seite wird die Verbindungsherstellung zu einem `DataSource`-Objekt dargestellt.

```
<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 erstellt und initialisiert das `DataSource` Objekt, bindet es an einen logischen Namen und registriert den Namen bei einem Java Naming and Directory Interface (JNDI) -Namensdienst. Der vollständige logische Name lautet `java:comp/env/`*`user-assigned-name`*. Sie müssen den vom Benutzer zugewiesenen Teil des Namens angeben, indem Sie benutzerdefinierte JSON-Attribute zur Stack-Konfiguration und zu den Bereitstellungsattributen hinzufügen, um das Attribut `['opsworks_java']['datasources']` zu definieren, wie nachfolgend beschrieben.

So stellen Sie eine JSP-Seite bereit, die eine Verbindung mit einer MySQL-Datenbank herstellt

1. Erstellen Sie einen Stack mit einer Java App Server-Ebene, fügen Sie jeder Ebene rund um die Uhr verfügbare Instanzen hinzu und starten Sie sie.
2. Fügen Sie eine Datenbankebene zum Stack hinzu. Die Details hängen davon ab, welche Datenbank Sie verwenden.

   Um eine MySQL-Instanz für das Beispiel zu verwenden, fügen Sie dem Stack eine MySQL-Ebene hinzu, fügen Sie der Ebene eine 24/7-Instanz hinzu und starten Sie sie.

   Um eine Amazon RDS (MySQL) -Instance für das Beispiel zu verwenden:

   - Geben Sie eine MySQL-Datenbank-Engine für die Instance an.
   - *Weisen Sie der* ***Instance die Sicherheitsgruppen AWS- OpsWorks -DB-Master-Server (security_group_id) und AWS- OpsWorks -Java-App-Server (security_group_id)*** *zu.* AWS OpsWorks Stacks erstellt diese Sicherheitsgruppen für Sie, wenn Sie Ihren ersten Stack in der Region erstellen.
   - Erstellen Sie eine Datenbank mit dem Namen `simplejspdb`.
   - Stellen Sie sicher, dass der Master-Benutzername und das Passwort nicht & oder andere Zeichen enthalten, die einen Tomcat-Fehler verursachen können.

     Insbesondere während des Starts muss Tomcat die Kontextdatei der Webanwendung analysieren. Diese XML-Datei enthält den Master-Benutzernamen und das Paswort. Wenn eine der Zeichenfolgen ein &-Zeichen enthält, interpretiert der XML-Parser dies als

manipuliertes XML-Objekt und löst eine Parsing-Ausnahme aus, sodass Tomcat nicht starten kann. Weitere Informationen über die Kontextdatei der Webanwendung finden Sie unter tomcat::context.

- Fügen Sie der Java App Server-Schicht einen MySQL-Treiber hinzu.
- Registrieren Sie die RDS-Instance mit Ihrem Stack.

Weitere Informationen zur Verwendung von Amazon RDS-Instances mit AWS OpsWorks Stacks finden Sie unterAmazon RDS-Serviceschicht.

3. Kopieren Sie den Beispiel-Code in eine Datei mit dem Namen `simplejspdb.jsp`, speichern Sie die Datei in einem Verzeichnis mit dem Namen `simplejspdb` und erstellen Sie ein `.zip`-Archiv des Verzeichnisses. Die Namen sind willkürlich. Sie können beliebige Datei- oder Verzeichnisnamen verwenden. Sie können auch andere Archivierungstypen einschließlich gzip, bzip2 oder Tarball verwenden. Um mehrere JSP-Seiten bereitzustellen, fügen Sie diese zu demselben Archiv hinzu. Weitere Informationen zur Bereitstellung von Anwendungen aus anderen Repository-Typen finden Sie unter Anwendungsquelle.
4. Laden Sie das Archiv in einen Amazon S3 S3-Bucket hoch und veröffentlichen Sie die Datei. Kopieren Sie die URL der Datei zur späteren Verwendung. Weitere Informationen zum Erstellen von Buckets und Hochladen von Dateien finden Sie unter Erste Schritte mit Amazon Simple Storage Service
5. Klicken Sie auf Hinzufügen einer Anwendung zum Stack und geben Sie folgende Einstellungen an:

   - Name (Name – `SimpleJSPDB`
   - App type – `Java`
   - Datenquellentyp — OpsWorks(für eine MySQL-Instance) oder RDS (für eine Amazon RDS-Instance).
   - Datenbank-Instance — Die zuvor erstellte MySQL-Instance, die normalerweise den Namen db-master1 (mysql) trägt, oder die Amazon RDS-Instance, die ***`DB_Instance_Name`*** (mysql) heißen wird.
   - Datenbankname – `simplejspdb`.
   - Repository-Typ – `Http Archive`
   - Repository-URL — die Amazon S3 S3-URL Ihrer Archivdatei.

Verwenden Sie die Standardwerte für die übrigen Einstellungen und klicken Sie anschließend auf Add App (App hinzufügen), um die Anwendung zu erstellen.

6. Fügen Sie die folgenden benutzerdefinierten JSON-Attribute zu den Stack-Konfigurationsattributen hinzu, wobei simplejspdb der Kurzname der Anwendung ist.

```
{
  "opsworks_java": {
    "datasources": {
      "simplejspdb": "jdbc/mydb"
    }
  }
}
```

   AWS OpsWorks Stacks verwendet diese Zuordnung, um eine Kontextdatei mit den erforderlichen Datenbankinformationen zu generieren.

   Weitere Informationen zum Hinzufügen benutzerdefinierter JSON-Attribute zu den Stack-Konfigurationsattributen finden Sie unter Nutzen eines benutzerdefinierten JSON-Objekts.

7. Stellen Sie die App auf der Java App Server-Instanz bereit.

Nun können Sie die Anwendung mit ihrer URL anzeigen. Eine Beschreibung zur Erstellung der URL finden Sie unter Bereitstellen einer JSP-Anwendung.

Die URL sollte in etwa wie folgt aussehen: `http://192.0.2.0/simplejspdb/simplejspdb.jsp`.

> **Note**
>
> Das `datasources`-Attribut kann mehrere Attribute enthalten. Jedes Attribut wird mit dem Namen des Kurznamens einer Anwendung bezeichnet und wird auf den entsprechenden, vom Benutzer zugewiesenen Teil eines logischen Namen gesetzt. Wenn Sie über mehrere Anwendungen verfügen, können Sie verschiedene logischen Namen verwenden. Dazu ist ein benutzerdefiniertes JSON-Objekt erforderlich, das in etwa wie folgt aussieht.
>
> ```
> {
> ```

```
  "opsworks_java": {
    "datasources": {
      "myjavaapp": "jdbc/myappdb",
      "simplejsp": "jdbc/myjspdb",
      ...
    }
  }
}
```

Node.js App Server AWS OpsWorks Stacks Layer

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Die App Server-Ebene von Node.js ist eine AWS OpsWorks Stacks-Ebene, die einen Blueprint für Instanzen bereitstellt, die als Node.js -Anwendungsserver fungieren. AWS OpsWorks Stacks installiert auch Express, sodass die Instanzen des Layers sowohl Standard- als auch Express-Anwendungen unterstützen.

Installation: Node.js wird installiert in `/usr/local/bin/node`.

Auf der Seite Add Layer (Ebene hinzufügen) stehen folgende Konfigurationsoptionen zur Verfügung:

Node.js-Version

Eine Liste der gegenwärtig unterstützten Versionen finden Sie unter AWS OpsWorks Stacks-Betriebssysteme.

Benutzerdefinierte Sicherheitsgruppen

Diese Einstellung wird angezeigt, wenn Sie Ihren Layern nicht automatisch eine integrierte AWS OpsWorks Stacks-Sicherheitsgruppe zuordnen möchten. Sie müssen die mit der Ebene zu verknüpfende Sicherheitsgruppe angeben. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

Elastic Load Balancer

Sie können den Instances des Layers einen Elastic Load Balancing Load Balancer zuordnen.

> **Important**
>
> Wenn Ihre Node.js-Anwendung SSL verwendet, empfehlen wir, SSLv3 zu deaktivieren, um die in CVE-2015-8027 beschriebenen Schwachstellen zu vermeiden. Um dies zu tun, müssen Sie Node.js version auf `0.12.9` festlegen.

Bereitstellen von Node.js-Anwendungen

Eine detaillierte Anleitung zur Implementierung einer einfachen Node.js-Anwendung für AWS OpsWorks Stacks und Bereitstellung für einen Stack finden Sie unter Erstellen Ihres ersten Node.js-Stacks. Normalerweise müssen Node.js-Anwendungen für AWS OpsWorks Stacks folgende Bedingungen erfüllen:

- Die Hauptdatei muss den Namen `server.js` haben und im Stammverzeichnis der Anwendung abgelegt sein.
- Im Stammverzeichnis von Express--Anwendungen muss die Datei `package.json` vorhanden sein.
- Standardmäßig muss die Anwendung über Port 80 (HTTP) oder Port 443 (HTTPS) kommunizieren.

  Es ist möglich, andere Ports abzuhören, aber die integrierte Sicherheitsgruppe der App Server-Schicht von Node.js, AWS- OpsWorks -NodeJS-App-Server, lässt eingehenden Benutzerverkehr nur zu den Ports 80, 443 und 22 (SSH) zu. Um eingehenden Benutzerverkehr zu anderen Ports zuzulassen, erstellen Sie eine Sicherheitsgruppe mit entsprechenden Regeln für eingehenden Datenverkehr und weisen Sie sie der App Server-Schicht Node.js zu. Ändern Sie keine Regeln für eingehenden Datenverkehr durch Bearbeiten der integrierten Sicherheitsgruppe. Jedes Mal, wenn Sie einen Stack erstellen, überschreibt AWS OpsWorks Stacks die integrierten Sicherheitsgruppen

mit den Standardeinstellungen, sodass alle von Ihnen vorgenommenen Änderungen verloren gehen.

> **Note**
>
> AWS OpsWorks Stacks setzt die Umgebungsvariable PORT auf 80 (Standard) oder 443 (wenn Sie SSL aktivieren), sodass Sie den folgenden Code verwenden können, um auf Anfragen zu warten.
>
> ```
> app.listen(process.env.PORT);
> ```

Wenn Sie eine App Node.js für die Unterstützung von SSL konfigurieren, müssen Sie den Schlüssel und die Zertifikate angeben. AWS OpsWorks Stacks platziert die Daten für jede Anwendungsserverinstanz wie folgt als separate Dateien im `/srv/www/`*`app_shortname`*`/shared/config` Verzeichnis.

- `ssl.crt`— das SSL-Zertifikat.
- `ssl.key`— der SSL-Schlüssel.
- `ssl.ca`— das Kettenzertifikat, falls Sie eines angegeben haben.

Ihre Anwendung kann die SSL-Schlüssel und Zertifikate aus diesen Dateien beziehen.

PHP-App-Server: AWS OpsWorks Stacks-Layer

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Die PHP App Server-Ebene ist eine AWS OpsWorks Stacks-Ebene, die einen Entwurf für Instanzen bereitstellt, die als PHP-Anwendungsserver fungieren. Die PHP App Server-Schicht basiert auf Apache2 mit `mod_php` und hat keine Standardkonfigurationsoptionen. Die PHP- und Apache-Version hängt davon ab, welches Betriebssystem Sie für die Ebene der Instances angeben.

Betriebssystem	PHP-Version	Apache-Version
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

Installation: AWS OpsWorks Stacks verwendet das Paketinstallationsprogramm der Instanz, um Apache2 und `mod_php` an ihren Standardspeicherorten zu installieren. Weitere Informationen zur Installation finden Sie unter Apache.

Auf der Seite Add Layer (Ebene hinzufügen) stehen folgende Konfigurationsoptionen zur Verfügung:

Benutzerdefinierte Sicherheitsgruppen

Diese Einstellung wird angezeigt, wenn Sie Ihren Layern nicht automatisch eine integrierte AWS OpsWorks Stacks-Sicherheitsgruppe zuordnen möchten. Sie müssen die mit der Ebene zu

verknüpfende Sicherheitsgruppe angeben. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

Elastic Load Balancer

Sie können den Instances des Layers einen Elastic Load Balancing Load Balancer zuordnen.

Mit einer benutzerdefinierten JSON- oder Attributdatei können einige Apache-Konfigurationseinstellungen geändert werden. Weitere Informationen finden Sie unter Überschreiben der Attribute. Eine Liste der überschreibbaren Apache-Attribute finden Sie unter apache2-Attribute.

Ein Beispiel für die Bereitstellung einer PHP-Anwendung einschließlich der Verbindung zu einer Backend-Datenbank finden Sie unter Erste Schritte mit Chef 11 Linux-Stacks.

Important

Wenn Ihre PHP-Anwendung SSL verwendet, empfehlen wir, SSLv3 zu deaktivieren, um die in CVE-2014-3566 beschriebenen Schwachstellen zu vermeiden. Dazu müssen Sie die Einstellung `SSLProtocol` in der Datei `ssl.conf` des Apache-Servers ändern. Weitere Informationen zum Ändern dieser Einstellung finden Sie unter Deaktivieren von SSLv3 für Apache-Server.

Rails App Server: AWS OpsWorks Stacks-Ebene

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Die Rails App Server-Schicht ist eine AWS OpsWorks Stacks-Schicht, die einen Entwurf für Instanzen bereitstellt, die als Rails-Anwendungsserver fungieren.

Installation: AWS OpsWorks Stacks verwendet das Paketinstallationsprogramm der Instanz, um die Serverpakete an ihren Standardspeicherorten zu installieren. Weitere Informationen zur Apache/Passenger-Installation finden Sie unter Phusion Passenger. Weitere Informationen zur Protokollierung finden Sie unter Log Files. Weitere Informationen zur Installation von Nginx finden Sie unter Unicorn.

Auf der Seite Add Layer (Ebene hinzufügen) stehen folgende Konfigurationsoptionen zur Verfügung, die alle optional sind.

Ruby-Version

Die von Ihren Anwendungen verwendete Ruby-Version. Der Standardwert ist 2.3.

Sie können auch Ihre bevorzugte Ruby-Version durch Überschreiben des `[:opsworks][:ruby_version]`-Attributs angeben.

> **Note**
>
> AWS OpsWorks Stacks installiert ein separates Ruby-Paket, das von Rezepten und dem Instanzagenten verwendet wird. Weitere Informationen finden Sie unter Ruby-Versionen.

Rails-Stack

Der standardmäßige Rails-Stack lautet Apache2 mit Phusion Passenger. Sie können auch Nginx mit Unicorn verwenden.

> **Note**
>
> Wenn Sie Ngnix und Unicorn verwenden, müssen Sie das Nginx- und Unicorn-Gem zur Gem-Datei Ihrer Anwendung hinzufügen, wie im folgenden Beispiel dargestellt:

```
source 'https://rubygems.org'
gem 'rails', '3.2.15'
...
# Use unicorn as the app server
```

```
gem 'unicorn'
...
```

Passenger-Version

Wenn Sie Apache2/Passenger angegeben haben, müssen Sie die Passenger-Version definieren. Der Standardwert ist 5.0.28.

RubyGems-Version

Die standardmäßige Rubygems-Version ist 2.5.1.

Bundler installieren und verwalten

Hier können Sie wählen, ob der Bundler installiert werden soll. Der Standardwert ist Yes.

Bundler-Version

Die standardmäßige Bundler-Version ist 1.12.5.

Benutzerdefinierte Sicherheitsgruppen

Diese Einstellung wird angezeigt, wenn Sie Ihren Layern nicht automatisch eine integrierte AWS OpsWorks Stacks-Sicherheitsgruppe zuordnen möchten. Sie müssen die mit der Ebene zu verknüpfende Sicherheitsgruppe angeben. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

Elastic Load Balancer

Sie können den Instances des Layers einen Elastic Load Balancing Load Balancer zuordnen.

Mit einer benutzerdefinierten JSON- oder Attributdatei können einige Konfigurationseinstellungen geändert werden. Weitere Informationen finden Sie unter Überschreiben der Attribute. Eine Liste der überschreibbaren Apache-, Nginx-, Phusion Passenger- und Unicorn-Attribute finden Sie unter Integrierte Rezeptbuchattribute.

Important

Wenn Ihre Ruby on Rails-Anwendung SSL verwendet, empfehlen wir, SSLv3 zu deaktivieren, um die in CVE-2014-3566 beschriebenen Schwachstellen zu vermeiden. Weitere Informationen finden Sie unter Deaktivieren von SSLv3 für Rails-Server.

Themen

- Deaktivieren von SSLv3 für Rails-Server
- Verbinden mit einer Datenbank
- Bereitstellen von Ruby on Rails-Anwendungen

Deaktivieren von SSLv3 für Rails-Server

Wenn Sie SSLv3 für Rails-Server deaktivieren möchten, aktualisieren Sie die Ruby Version (Ruby-Version)-Einstellung der Ebene auf 2.1 oder höher, sodass Ruby 2.1.4 oder höher als die von den Anwendungen verwendete Version installiert wird.

- Aktualisieren Sie die Ruby Version--Einstellung der Ebene auf 2.1 oder höher.
- Aktualisieren Sie die Konfigurationsdatei für Ihren Rails-Stack folgendermaßen.

Apache mit Phusion Passenger

Aktualisieren Sie die `SSLProtocol`-Einstellung in der `ssl.conf`-Datei des Apache-Servers, wie in Deaktivieren von SSLv3 für Apache-Server beschrieben.

Nginx mit Unicorn

Fügen Sie eine explizite `ssl_protocols`-Richtlinie zur `nginx.conf`-Datei des Nginx-Servers hinzu. Um SSLv3 zu deaktivieren, überschreiben Sie die `nginx.conf.erb` Vorlagendatei des integrierten Nginx-Kochbuchs, die die Setup-Rezepte der Rails App Server-Schicht zum Erstellen verwendenn`ginx.conf`, und fügen Sie die folgende Direktive hinzu:

```
ssl_protocols TLSv1.2;
```

Weitere Informationen zum Konfigurieren von `nginx.conf` finden Sie unter Configuring HTTPS servers. Weitere Informationen zum Überschreiben einer integrierten Vorlage finden Sie unter Verwenden von benutzerdefinierten Vorlagen.

Verbinden mit einer Datenbank

Wenn Sie eine App bereitstellen, erstellt AWS OpsWorks Stacks eine neue `database.yml` Datei mit Informationen aus den Attributen der App. `deploy` Wenn Sie eine MySQL- oder Amazon RDS-Instance an die App anhängen, fügt AWS OpsWorks Stacks die Verbindungsinformationen zu den

`deploy` Attributen hinzu, sodass sie `database.yml` automatisch die richtigen Verbindungsdaten enthalten.

Wenn eine App keine angehängte Datenbank hat, fügt AWS OpsWorks Stacks den `deploy` Attributen standardmäßig keine Verbindungsinformationen hinzu und erstellt auch keine. `database.yml` Wenn Sie eine andere Datenbank verwenden möchten, können Sie mit einem benutzerdefinierten JSON-Objekt Verbindungsdaten enthaltende Datenbank-Attribute zu den deploy-Attributen der Anwendung hinzufügen. Die Attribute befinden sich alle unter`["deploy"]["`*`appshortname`*`"]["database"]`, wobei *`appshortname`* der Kurzname der App ist, den AWS OpsWorks Stacks aus dem App-Namen generiert. Die in der benutzerdefinierten JSON-Datei angegebenen Werte überschreiben sämtliche Standardeinstellungen. Weitere Informationen finden Sie unter Hinzufügen von Apps.

AWS OpsWorks Stacks integriert die folgenden `[:...][:database]`Attributwerte in. `database.yml` Die erforderlichen Attribute hängen von der jeweiligen Datenbank ab, aber Sie müssen über ein `host` Attribut verfügen, da AWS OpsWorks Stacks sonst nichts erstellt. `database.yml`

- `[:adapter] (String)`— Der Datenbankadapter, wie z. `mysql`
- `[:database]`(String) — Der Datenbankname.
- `[:encoding]`(String) — Die Kodierung, auf die normalerweise eingestellt ist`utf8`.
- `[:host]`(String) — Die Host-URL, z. `railsexample.cdlqlk5uwd0k.us-west-2.rds.amazonaws.com` B.
- `[:reconnect]`(Boolean) — Gibt an, ob die Anwendung erneut eine Verbindung herstellen soll, wenn die Verbindung nicht mehr besteht.
- `[:password]`(String) — Das Datenbankkennwort.
- `[:port]` (Zahl). — Die Portnummer der Datenbank. Mit diesem Attribut können Sie die Standard-Portnummer überschreiben, der vom Adapter vorgegeben wird.
- `[:username]`(String) — Der Datenbankbenutzername.

Das folgende Beispiel zeigt ein benutzerdefiniertes JSON-Objekt für eine Anwendung mit dem Kurznamen myapp.

```
{
  "deploy" : {
    "myapp" : {
```

```
      "database" : {
        "adapter" : "adapter",
        "database" : "databasename",
        "host" : "host",
        "password" : "password",
        "port" : portnumber
        "reconnect" : true/false,
        "username" : "username"
      }
    }
  }
}
```

Weitere Informationen zur Definition eines benutzerdefinierten JSON-Objekts finden Sie unter Nutzen eines benutzerdefinierten JSON-Objekts. Um die Vorlage zur Erstellung von `database.yml` (`database.yml.erb`) zu sehen, gehen Sie zu dem integrierten Rezeptbuch-Repository.

Bereitstellen von Ruby on Rails-Anwendungen

Sie können Ruby on Rails-Anwendungen aus einem unterstützten Repository bereitstellen. Im folgenden Beispiel wird erläutert, wie eine beispielhafte Ruby on Rails-Anwendung für einen Server bereitgestellt wird, auf dem ein Apache/Passenger-Rails-Stack ausgeführt wird. Der Beispielcode ist in einem öffentlichen GitHub Repository gespeichert, aber das grundlegende Verfahren ist dasselbe für die anderen unterstützten Repositorys. Weitere Informationen zum Erstellen und Bereitstellen von Anwendungen finden Sie unter Apps. Den Code des Beispiels, der ausführliche Kommentare enthält, finden Sie unter https://github.com/awslabs/ opsworks-demo-rails-photo -share-app.

Um eine Ruby on Rails-App aus einem Repository bereitzustellen GitHub

1. Erstellen Sie einen Stack mit einer Rails-App Server-Ebene mit Apache/Passenger als Rails-Stack, fügen Sie der Ebene eine 24/7-Instanz hinzu und starten Sie sie.
2. Wenn die Instance online ist, fügen Sie eine Anwendung zum Stack hinzu und geben Sie folgenden Einstellungen an:

   - Name – Beliebiger Name, im Beispiel wird `PhotoPoll` verwendet.

     AWS OpsWorks Stacks verwendet diesen Namen zu Anzeigezwecken und generiert einen Kurznamen für den internen Gebrauch und zur Identifizierung der App in der Stackkonfiguration und den Bereitstellungsattributen. Der PhotoPoll Kurzname lautet beispielsweise photopoll.

- App type – Ruby on Rails.
- Rails environment – Die verfügbaren Umgebungen werden von der Anwendung bestimmt.

  Die Beispiel-Anwendung hat drei Umgebungen: **`development`**, **`test`**, und **`production`**. In diesem Beispiel geben Sie für die Umgebung **`development`** an. Im Beispiel-Code sind weitere Beschreibungen für jede Umgebung enthalten.
- Repository-Typ — Jeder der unterstützten Repository-Typen. Geben Sie `Git` für dieses Beispiel an.
- Repository URL – Repository, aus dem der Code bereitzustellen ist.

  In diesem Beispiel geben Sie für die URL **`git://github.com/awslabs/opsworks-demo-rails-photo-share-app`** an.

Verwenden Sie die Standardwerte für die übrigen Einstellungen und klicken Sie anschließend auf Add App (App hinzufügen), um die Anwendung zu erstellen.

3. Stellen Sie die App auf der Rails App Server-Instanz bereit.
4. Wenn die Bereitstellung abgeschlossen ist, gehen Sie zur Seite Instanzen und klicken Sie auf die öffentliche IP-Adresse der Rails App Server-Instanz. Sie sollten Folgendes sehen:

Statischer Webserver: AWS OpsWorks Stacks Layer

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Die statische Webserver-Ebene ist eine AWS OpsWorks Stacks-Ebene, die eine Vorlage für Instanzen zur Bereitstellung statischer HTML-Seiten bereitstellt, zu denen auch clientseitiges Scripting gehören kann. Diese Ebene basiert auf Nginx.

Installation: Nginx wird installiert in `/usr/sbin/nginx`.

Auf der Seite Add Layer (Ebene hinzufügen) stehen folgende Konfigurationsoptionen zur Verfügung:

Benutzerdefinierte Sicherheitsgruppen

Diese Einstellung wird angezeigt, wenn Sie Ihren Layern nicht automatisch eine integrierte AWS OpsWorks Stacks-Sicherheitsgruppe zuordnen möchten. Sie müssen die mit der Ebene zu verknüpfende Sicherheitsgruppe angeben. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

Elastic Load Balancer

Sie können den Instances des Layers einen Elastic Load Balancing Load Balancer zuordnen.

Mit einer benutzerdefinierten JSON- oder Attributdatei können einige Nginx-Konfigurationseinstellungen geändert werden. Weitere Informationen finden Sie unter Überschreiben der Attribute. Eine Liste der überschreibbaren Apache-Attribute finden Sie unter nginx-Attribute.

Important

Wenn Ihre Webanwendung SSL verwendet, empfehlen wir, SSLv3 zu deaktivieren, um die in CVE-2014-3566 beschriebenen Schwachstellen zu vermeiden.
Um SSLv3 zu deaktivieren, müssen Sie die Datei `nginx.conf` des Nginx-Servers ändern. Überschreiben Sie dazu die `nginx.conf.erb` Vorlagendatei des integrierten Nginx-Kochbuchs, die die Setup-Rezepte der Rails App Server-Schicht zum Erstellen verwendenn`ginx.conf`, und fügen Sie die folgende Direktive hinzu:

```
ssl_protocols TLSv1.2;
```

Weitere Informationen zum Konfigurieren von `nginx.conf` finden Sie unter Configuring HTTPS servers. Weitere Informationen zum Überschreiben einer integrierten Vorlage finden Sie unter Verwenden von benutzerdefinierten Vorlagen.

Referenz zur ECS-Clusterschicht

Note

Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Eine ECS-Cluster-Schicht stellt einen Amazon Elastic Container Service (Amazon ECS) -Cluster dar und vereinfacht die Clusterverwaltung.

Short name (Kurzname): ecs-cluster

Kompatibilität: Eine Amazon ECS-Serviceschicht ist nur mit benutzerdefinierten Ebenen kompatibel

Offene Ports: Der ECS-Cluster ermöglicht den öffentlichen Zugriff auf Port 22 (SSH)

Autoassign Elastic IP addresses (Elastic IP-Adressen automatisch zuweisen): Standardmäßig deaktiviert

Default EBS volume (Standard-EBS-Volume): Nein

Standard-Sicherheitsgruppe: AWS- OpsWorks -ECS-Cluster

Konfiguration: Um eine ECS-Cluster-Schicht zu konfigurieren, müssen Sie Folgendes angeben:

- Ob den Container-Instances öffentliche IP-Adressen oder Elastic IP-Adressen zugewiesen werden sollen
- Das Instance-Profil für die Container-Instances

Setup recipes (Einrichtungsrezepte):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- vermeiden
- ebs
- opsworks_ganglia::client
- opsworks_ecs::setup

Configure recipes (Konfigurationsrezepte):

- opsworks_ganglia::configure-client
- ssh_users
- mysql::client
- agent_version
- opsworks_ecs::configure

Deploy recipes (Bereitstellungsrezepte):

- deploy::default
- opsworks_ecs::deploy

Undeploy recipes (Bereitstellung von Rezepten aufheben):

- opsworks_ecs::undeploy

Shutdown recipes (Shutdown-Rezepte):

- opsworks_shutdown::default
- opsworks_ecs::shutdown

Installation (Installation):

- AWS OpsWorks Stacks verwendet das Paketinstallationsprogramm der Instanz, um Docker an seinen Standardspeicherorten zu installieren
- Das Chef-Protokoll für das Setup-Ereignis vermerkt, ob der Amazon ECS-Agent erfolgreich installiert wurde. Andernfalls enthalten die von AWS OpsWorks Stacks bereitgestellten Protokolle keine Amazon ECS-Fehlerprotokollinformationen. Weitere Informationen zur Behandlung von Amazon ECS-Fehlern finden Sie unter Amazon ECS-Fehlerbehebung.

Referenz für benutzerdefinierte Layer

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn die Standard-Layer nicht Ihren Anforderungen entsprechen, können Sie einen benutzerdefinierten Layer erstellen. Ein Stack kann mehrere benutzerdefinierte Layer aufweisen. Standardmäßig führt der benutzerdefinierte Layer bestimmte standardmäßige Rezepte aus, die grundlegende Funktionen unterstützen. Sie implementieren dann die primäre Funktionalität des Layers durch Implementieren einer Reihe benutzerdefinierter Chef-Rezepte für jedes der entsprechenden Lebenszyklusereignisse, zum Einrichten und Konfigurieren der Software des Layers usw. Benutzerdefinierte Rezepte folgen den AWS OpsWorks Standard-Stacks-Rezepten für jedes Event.

Short name (Kurzname): Benutzerdefiniert. Jeder benutzerdefinierte Layer in einem Stack muss einen unterschiedlichen Kurznamen haben.

Open ports (Offene Ports): Standardmäßig öffnet ein benutzerdefinierter Server-Layer den öffentlichen Zugriff auf die Ports 22 (SSH), 80 (HTTP), 443 (HTTPS) und alle Ports der Rails- und PHP-Anwendungsserver-Layer des Stacks.

Autoassign Elastic IP Addresses (Elastic IP-Adresse automatisch zuweisen): Standardmäßig deaktiviert

Default EBS volume (Standard-EBS-Volume): Nein

Standard-Sicherheitsgruppe: AWS- OpsWorks -Custom-Server

Compatibility (Kompatibilität): Benutzerdefinierte Layer sind mit folgenden Layern kompatibel: benutzerdefiniert, DB-Master, LB, Memcached, Monitoring-Master, Nodejs-App, PHP-App, Rails-App und Web.

Configuration (Konfiguration): Um einen benutzerdefinierten Layer zu konfigurieren, müssen Sie Folgendes angeben:

- Den Namen des Layers
- Den Kurznamen des Layers, der den Layer in Chef-Rezepte angibt und nur Buchstaben von A bis Z und Zahlen aufweist.

Bei Linux-Stacks verwendet der benutzerdefinierte Layer die folgenden Rezepte.

Setup recipes (Einrichtungsrezepte):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- vermeiden
- ebs
- opsworks_ganglia::client

Configure recipes (Konfigurationsrezepte):

- opsworks_ganglia::configure-client
- ssh_users

- agent_version

Deploy recipes (Bereitstellungsrezepte):

- deploy::default

Shutdown recipes (Shutdown-Rezepte):

- opsworks_shutdown::default

Referenz für andere Layer

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks unterstützt auch die folgenden Ebenen.

Themen

- Referenz zur Ganglien-Ebene
- Memcached-Layer-Referenz

Referenz zur Ganglien-Ebene

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Eine Ganglia-Ebene unterstützt Ganglia, ein verteiltes Überwachungssystem, das die Speicherung und Visualisierung von Instanzmetriken verwaltet. Es wurde zur Zusammenarbeit mit hierarchischen Instance-Topologien entwickelt, was es besonders nützlich für Instance-Gruppen macht. Ganglia hat zwei grundlegende Komponenten:

- Einen Client mit geringem Overhead, der auf jeder Instance im Stack installiert wird und Metriken an den Master sendet.
- Ein Master, der Metriken von den Clients sammelt und sie auf einem Amazon EBS-Volume speichert. Außerdem werden die Metriken auf einer Webseite angezeigt.

AWS OpsWorks Stacks hat auf jeder Instanz, die es verwaltet, einen Ganglia-Monitoring-Agenten. Wenn Sie Ihrem Stack eine Ganglia-Ebene hinzufügen und diese starten, melden die Ganglia-Agenten auf jeder Instanz Metriken an die Ganglia-Instanz. Um Ganglia zu verwenden, fügen Sie dem Stack eine Ganglia-Ebene mit einer Instanz hinzu. Sie haben Zugriff auf die Daten, indem Sie sich beim Ganglia-Backend unter der IP-Adresse des Masters anmelden. Sie können zusätzliche Metrikdefinitionen bereitstellen, indem Sie Chef-Rezepte schreiben.

Short name (Kurzname): monitoring-master

Kompatibilität: Eine Ganglia-Schicht ist mit den folgenden Ebenen kompatibel: custom, db-master, memcached, php-app, rails-app.

Open ports (Offene Ports): Load-Balancer ermöglicht den öffentlichen Zugriff auf die Ports 22 (SSH), 80 (HTTP) und 443 (HTTPS).

Autoassign Elastic IP addresses (Elastic IP-Adressen automatisch zuweisen): Standardmäßig deaktiviert

Default EBS volume (Standard-EBS-Volume): Ja, bei `/vol/ganglia`

Standard-Sicherheitsgruppe: AWS- OpsWorks -Monitoring-Master-Server

Konfiguration: Um eine Ganglia-Schicht zu konfigurieren, müssen Sie Folgendes angeben:

- Die URI, die den Zugriff auf die Überwachungsdiagramme bereitstellt. Der Standardwert ist http://*dnsName /ganglia, wobei dnsName der DNS-Name der Ganglia-Instanz* ist.
- Den Benutzernamen und das Passwort zur Steuerung des Zugriffs auf die Überwachungsstatistiken.

Setup recipes (Einrichtungsrezepte):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- vermeiden
- ebs
- opsworks_ganglia::client
- opsworks_ganglia::server

Configure recipes (Konfigurationsrezepte):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- opsworks_ganglia::configure-server

Deploy recipes (Bereitstellungsrezepte):

- deploy::default
- opsworks_ganglia::configure-server
- opsworks_ganglia::deploy

Shutdown recipes (Shutdown-Rezepte):

- opsworks_shutdown::default
- apache2::stop

Installation (Installation):

- Der Ganglia-Client wird installiert unter: `/etc/ganglia`.
- Der Ganglia-Web-Frontend wird installiert unter: `/usr/share/ganglia-webfrontend`.
- Der Ganglia-Logtailer wird installiert unter: `/usr/share/ganglia-logtailer`.

Memcached-Layer-Referenz

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebene steht nur für Linux-basierte Stacks zur Verfügung.

Memcached ist ein verteiltes Speicher-Caching-System für beliebige Daten. Es beschleunigt Websites, indem Zeichenfolgen und Objekte als Schlüssel und Werte im RAM gespeichert werden, um die Anzahl der Male, die eine externe Datenquelle gelesen werden muss, zu reduzieren.

Um Memcached in einem Stack zu verwenden, erstellen Sie eine Memcached-Ebene und fügen Sie eine oder mehrere Instanzen hinzu, die als Memcached-Server fungieren. Die Instanzen installieren Memcached automatisch, und die anderen Instanzen des Stacks können auf die Memcached-Server zugreifen und diese verwenden. Wenn Sie einen Rails-App-Server-Layer verwenden, platziert AWS OpsWorks Stacks automatisch eine `memcached.yml` Konfigurationsdatei im Konfigurationsverzeichnis jeder Instanz in der Ebene. Sie können den Memcached-Server und die Portnummer aus dieser Datei abrufen.

Short name (Kurzname): memcached

Kompatibilität: Eine Memcached-Schicht ist mit den folgenden Ebenen kompatibel: custom, db-Master, lb, Monitoring-Master, Nodejs-App, PHP-App, Rails-App und Web.

Offene Ports: Eine Memcached-Schicht ermöglicht den öffentlichen Zugriff auf Port 22 (SSH) und alle Ports von den Webservern, benutzerdefinierten Servern und den Anwendungsservern Rails, PHP und Node.js des Stacks.

Autoassign Elastic IP addresses (Elastic IP-Adressen automatisch zuweisen): Standardmäßig deaktiviert

Default EBS volume (Standard-EBS-Volume): Nein

Standard-Sicherheitsgruppe: AWS- OpsWorks -Memcached-Server

Um einen Memcached-Layer zu konfigurieren, müssen Sie die Cache-Größe in MB angeben.

Setup recipes (Einrichtungsrezepte):

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- vermeiden
- ebs
- opsworks_ganglia::client
- memcached

Configure recipes (Konfigurationsrezepte):

- opsworks_ganglia::configure-client
- ssh_users
- agent_version

Deploy recipes (Bereitstellungsrezepte):

- deploy::default

Shutdown recipes (Shutdown-Rezepte):

- opsworks_shutdown::default
- memcached::stop

Installation (Installation):

- AWS OpsWorks Stacks verwendet das Paketinstallationsprogramm der Instanz, um Memcached und seine Protokolldateien an ihren Standardspeicherorten zu installieren.

Andere Layer

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Ebenen sind nur für Chef 11 und frühere Linux-basierte Stacks verfügbar.

AWS OpsWorks Stacks unterstützt auch die Ebenen Ganglia und Memcached.

Themen

Ganglien-Schicht

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Dieser Layer steht nur für Chef 11 und niedrigere Linux-basierte Stacks zur Verfügung.

AWS OpsWorks Stacks sendet all Ihre Instance- und Volume-Metriken an Amazon CloudWatch, sodass Sie auf einfache Weise Grafiken anzeigen und Alarme einrichten können, um Ihnen bei der Fehlerbehebung zu helfen und automatisierte Maßnahmen auf der Grundlage des Status Ihrer Ressourcen zu ergreifen. Sie können die Ganglia AWS OpsWorks Stacks-Ebene auch für zusätzliche Optionen zur Anwendungsüberwachung verwenden, z. B. für das Speichern Ihrer ausgewählten Metriken.

Die Ganglia-Schicht ist eine Blaupause für eine Instanz, die Ihren Stack mithilfe von Ganglia Distributed Monitoring überwacht. Ein Stack hat normalerweise nur eine Ganglia-Instanz. Die Ganglia-Schicht umfasst die folgenden optionalen Konfigurationseinstellungen:

Ganglia URL (URL für Ganglia)

Den URL-Pfad der Statistik. Die komplette URL lautet http://*DNSNameURLPath*, wobei *DNSName* der zugeordnete DNS-Name der Instance ist. Der Standardwert für *URLPath* ist „/ganglia“, was etwa folgendem entspricht: http://ec2-54-245-151-7.us-west-2.compute.amazonaws.com/ganglia.

Ganglia user name (Benutzername für Ganglia)

Ein Benutzername für die Statistik-Webseite. Sie müssen den Benutzernamen angeben, wenn Sie die Seite aufrufen. Der Standardwert ist „opsworks“.

Ganglia password (Passwort für Ganglia)

Ein Passwort, mit dem der Zugriff auf die Statistik-Webseite kontrolliert wird. Zum Anzeigen der Seite müssen Sie das Passwort eingeben. Der Standardwert ist eine zufällig erstellte Zeichenfolge.

> **Note**
>
> Notieren Sie sich das Passwort für die spätere Verwendung. In AWS OpsWorks Stacks können Sie das Passwort nicht anzeigen, nachdem Sie die Ebene erstellt haben. Sie

können jedoch das Passwort aktualisieren, indem Sie die Bearbeitungsseite des Layers aufrufen und auf Update password (Passwort aktualisieren) klicken.

Benutzerdefinierte Sicherheitsgruppen

Diese Einstellung wird angezeigt, wenn Sie Ihren Layern nicht automatisch eine integrierte AWS OpsWorks Stacks-Sicherheitsgruppe zuordnen möchten. Sie müssen die mit der Ebene zu verknüpfende Sicherheitsgruppe angeben. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

Elastic Load Balancer

Sie können den Instances des Layers einen Elastic Load Balancing Load Balancer zuordnen.

Important

Wenn Ihr Stack eine Ganglia-Schicht enthält, empfehlen wir Ihnen, SSLv3 nach Möglichkeit für diese Schicht zu deaktivieren, um die in CVE-2014-3566 beschriebenen Sicherheitslücken zu beheben. Dazu müssen Sie die Vorlage `ssl.conf.erb` des Apache-Servers überschreiben, um die `SSLProtocol`-Einstellung zu ändern. Details hierzu finden Sie unter Deaktivieren von SSLv3 für Apache-Server.

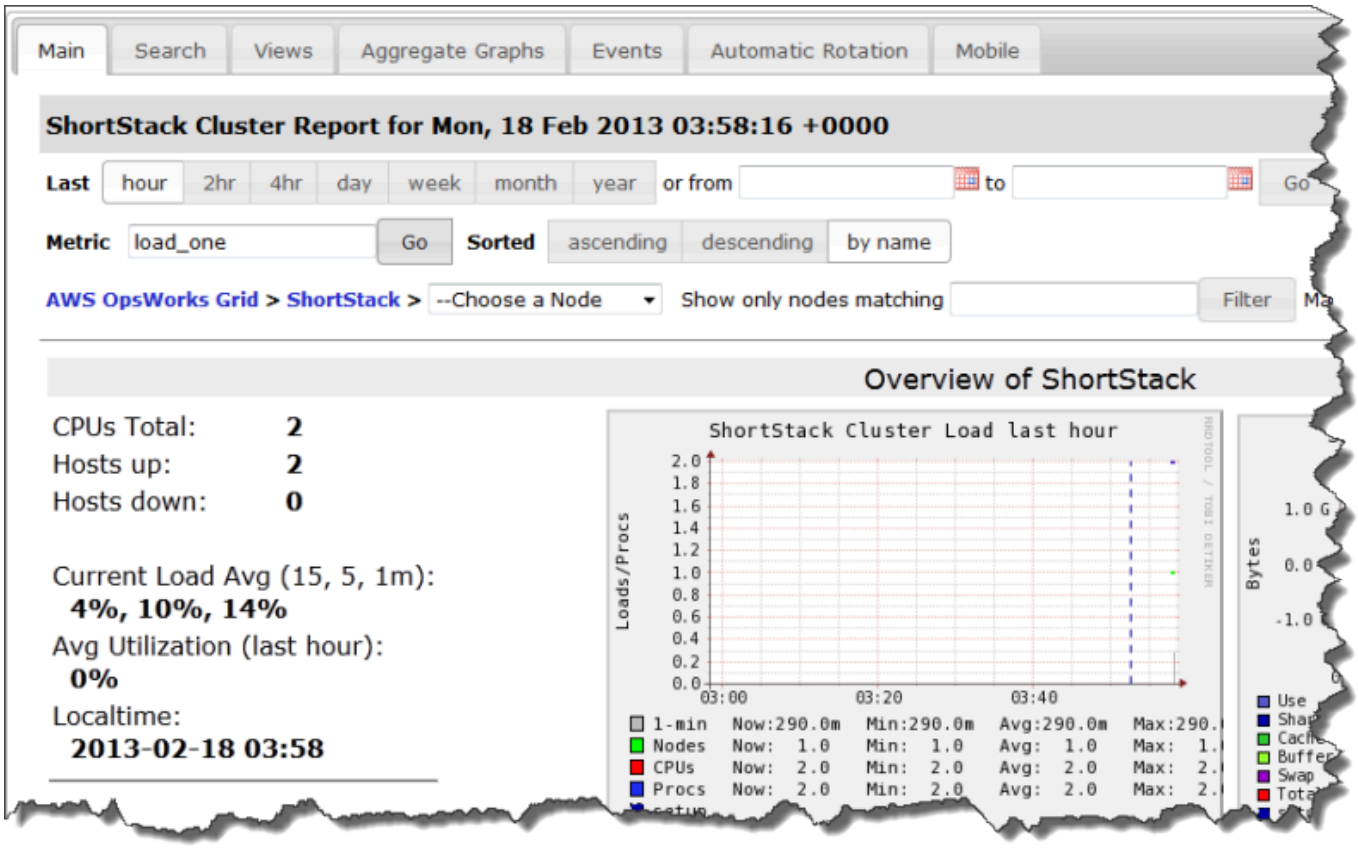
Sehen Sie sich die Ganglia-Statistik an

AWS OpsWorks Stacks-Rezepte installieren auf jeder Instanz einen Ganglia-Client mit geringem Overhead. Wenn Ihr Stack eine Ganglia-Ebene enthält, beginnt der Ganglia-Client automatisch mit der Berichterstattung an die Ganglia, sobald die Instanz online ist. Ganglia verwendet die Kundendaten, um eine Vielzahl von Statistiken zu berechnen, und zeigt die Ergebnisse grafisch auf seiner Statistik-Webseite an.

So zeigen Sie Ganglia-Statistiken an

1. Klicken Sie auf der Seite „Ebenen“ auf Ganglia, um die Detailseite der Ebene zu öffnen.
2. Klicken Sie im Navigationsbereich auf Instances. Klicken Sie unter Ganglia auf den Instanznamen.
3. Kopieren Sie den Public DNS-Namen der Instance.

4. Verwenden Sie den DNS-Namen, um die Statistik-URL zu erstellen, die etwa wie folgt aussieht: http://ec2-54-245-151-7.us-west-2.compute.amazonaws.com/ganglia.
5. Kopieren Sie die vollständige URL in Ihren Browser, navigieren Sie zur Seite und geben Sie den Benutzernamen und das Passwort für Ganglia ein, um die Seite anzuzeigen. Ein Beispiel folgt.



Memcached

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Dieser Layer ist nur für Chef 11 und frühere Linux-basierte Stacks verfügbar.

Ein Memcached-Layer ist ein AWS OpsWorks Stacks-Layer, der eine Blaupause für Instances bereitstellt, die als Memcached-Server fungieren — ein verteiltes Speicher-Caching-System für beliebige Daten. Die Memcached-Schicht umfasst die folgenden Konfigurationseinstellungen.

Allocated memory (MB) (Zugewiesener Speicher (MB))

(Optional) Den Cache-Speicher (in MB) für jede Layer-Instance. Der Standardwert ist 512 MB.

Benutzerdefinierte Sicherheitsgruppen

Diese Einstellung wird angezeigt, wenn Sie Ihren Layern nicht automatisch eine integrierte AWS OpsWorks Stacks-Sicherheitsgruppe zuordnen möchten. Sie müssen die mit der Ebene zu verknüpfende Sicherheitsgruppe angeben. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

Bestandteile eines Rezeptbuchs

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Ein Rezeptbuch besteht normalerweise aus den folgenden Basiskomponenten:

- Attributdateien enthalten eine Reihe von Attributen mit Werten, die von Rezepten und Vorlagen verwendet werden.
- Vorlagendateien sind Vorlagen, auf Basis derer Rezepte andere Dateien wie Konfigurationsdateien erstellen.

Mit Vorlagendateien können Sie in der Regel die Konfigurationsdatei ändern, indem Sie Attribute überschreiben — was ohne Berührung mit dem Kochbuch geschehen kann —, anstatt eine Konfigurationsdatei neu zu schreiben. In der Praxis sollten Sie sämtliche Änderungen an Konfigurationsdateien auf einer Instance, auch wenn sie nur geringfügig sind, mithilfe von Vorlagendateien vornehmen.

- Rezeptdateien sind Ruby-Anwendungen, in denen sämtliche Informationen zur Systemkonfiguration definiert werden, einschließlich dem Erstellen und Konfigurieren von Verzeichnissen, dem Installieren und Konfigurieren von Softwarepaketen, dem Starten von Services usw.

Rezeptbücher müssen nicht aus allen drei Komponenten bestehen. Für einfachere Anpassungen benötigen Sie nur Attribut- oder Vorlagendateien. Darüber hinaus können Rezeptbücher noch weitere Dateitypen wie Definitionen oder Spezifikationen enthalten.

In diesem Abschnitt werden die drei Standardkomponenten von Rezeptbüchern beschrieben. Weitere Informationen insbesondere auch zum Implementieren von Rezepten finden Sie unter Opscode.

Themen

- Attribute
- Vorlagen
- Rezepte

Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Rezepte und Vorlagen sind von einer ganzen Reihe von Werten, beispielsweise Konfigurationseinstellungen, abhängig. Statt diese Werte direkt in Rezepten oder Vorlagen fest

zu programmieren, können Sie eine Attributdatei mit Attributen für die benötigten Werte erstellen. Diese Attribute verwenden Sie dann statt der tatsächlichen Werte in Rezepten und Vorlagen. Der Vorteil dieser Methode besteht darin, dass Sie Werte überschreiben können, ohne Änderungen am Rezeptbuch vornehmen zu müssen. Daher sollten Sie folgende Arten von Werten stets durch Attribute definieren:

- Werte, die sich je nach Stack oder im Laufe der Zeit ändern können, z. B. Benutzernamen

  Wenn Sie solche Werte fest programmieren, müssen Sie jedes Mal, wenn sich ein Wert ändert, das Rezept bzw. die Vorlage ändern. Wenn Sie diese Werte jedoch durch Attribute definieren, können Sie dieselben Rezeptbücher auf allen Stacks verwenden und müssen nur die entsprechenden Attribute überschreiben.
- Sensible Werte wie Passwörter oder geheime Schlüssel

  Wenn Sie sensible Werte in Rezeptbüchern speichern, ist die Gefahr höher, dass diese offengelegt werden. Definieren Sie stattdessen Attribute mit Platzhalterwerten und überschreiben Sie diese mit den tatsächlichen Werten. Am einfachsten überschreiben Sie solche Attribute mit benutzerdefinierter JSON. Weitere Informationen finden Sie unter Nutzen eines benutzerdefinierten JSON-Objekts.

Weitere Informationen zu Attributen und wie diese überschrieben werden können, finden Sie unter Überschreiben der Attribute.

Das nachfolgende Beispiel ist ein Auszug aus einer Beispielattributdatei.

```
...
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 definiert Attribute mithilfe der folgenden Syntax:

```
node.type["attribute"]["subattribute"]["..."]=value
```

Sie können Doppelpunkte (:) wie folgt verwenden:

```
node.type[:attribute][:subattribute][:...]=value
```

Eine Attributdefinition besteht aus den folgenden Komponenten:

`node.`

Das Präfix `node.` ist optional und wird in der Regel ausgelassen, wie in diesem Beispiel gezeigt.

`type`

Der Typ bestimmt, ob das Attribut überschrieben werden kann. AWS OpsWorks Stacks-Attribute verwenden in der Regel einen der folgenden Typen:

- `default` wird am häufigsten verwendet, da Attribute dieses Typs überschrieben werden können.
- `normal`definiert ein Attribut, das einen der standardmäßigen AWS OpsWorks Stacks-Attributwerte überschreibt.

> **Note**
>
> Chef unterstützt zusätzliche Typen, die für AWS OpsWorks Stacks nicht erforderlich sind, aber für Ihr Projekt nützlich sein könnten. Weitere Informationen zu Attributen finden Sie unter About Attributes.

`attribute name`

Der Attributname folgt der Standard-Chef-Knotensyntax, `[:attribute][:subattribute][...]`. Sie können für Attribute beliebige Namen verwenden. Allerdings werden benutzerdefinierte Rezeptbuchattribute, wie in Überschreiben der Attribute erläutert, mit dem Knotenobjekt der Instance zusammengeführt, zusammen mit den Attributen von der Stack-Konfiguration und den Bereitstellungsattributen sowie dem Ohai-Tool von Chef. Häufig verwendete Konfigurationsnamen wie port oder user werden in zahlreichen Rezeptbüchern verwendet.

Um Namensüberschneidungen zu vermeiden, sollten Sie qualifizierte Attributnamen mit mindestens zwei Elementen erstellen, wie in diesem Beispiel gezeigt. Das erste Element sollte eindeutig sein und bezieht sich in der Regel auf einen Produktnamen wie Apache. Ihm folgen ein oder mehrere Unterattribute, die den eigentlichen Wert festlegen, z. B. `[:user]` oder `[:port]`. Sie können beliebig viele Unterattribute verwenden; dies hängt auch von der Struktur Ihres Projekts ab.

`value`

Ein Attribut kann folgende Werttypen aufweisen:

- Eine Zeichenfolge, z. B. `default[:apache][:keepalive] = 'Off'`
- Eine Zahl (ohne Anführungszeichen), z. B. `default[:apache][:timeout] = 120`
- Ein boolescher Wert, entweder `true` oder `false` (ohne Anführungszeichen)
- Eine Liste mit Werten, z. B. `default[:apache][:listen_ports] = [ '80','443' ]`

Die Attributdatei ist eine Ruby-Anwendung, daher können Sie auch Knotensyntax und logische Operatoren verwenden, um Werte basierend auf anderen Attributen zuzuweisen. Weitere Informationen zur Definition von Attributen finden Sie unter About Attributes. Beispiele für funktionierende Attributdateien finden Sie in den integrierten Kochbüchern von AWS OpsWorks Stacks unter https://github.com/aws/opsworks-cookbooks.

Vorlagen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Viele Pakete lassen sich über eine Konfigurationsdatei konfigurieren, die im passenden Verzeichnis gespeichert wird. Sie können eine Konfigurationsdatei in einem Rezeptbuch speichern und in das entsprechende Verzeichnis kopieren. Eine flexiblere Methode ist es jedoch, die Konfigurationsdatei mithilfe eines Rezepts aus einer Vorlage zu erstellen. Ein Vorteil von Vorlagen besteht darin, dass Sie die Werte der Vorlage mithilfe von Attributen festlegen können. So lassen sich beispielsweise

Konfigurationsdateien bearbeiten, ohne dass Sie Änderungen am Rezeptbuch vornehmen müssen, indem Sie einfach mit benutzerdefinierter JSON die entsprechenden Attributwerte überschreiben.

Eine Vorlage hat im Wesentlichen denselben Inhalt und dieselbe Struktur wie die zugehörige Datei. Hier sehen Sie eine Beispieldatei, `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>
...
```

Das folgende Beispiel besteht aus der Datei `httpd.conf` für eine Ubuntu-Instance:

```
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>
...
```

Ein Großteil des Inhalts der Vorlage wurde einfach aus der Vorlage in die Datei `httpd.conf` kopiert. `<%= ... %>`-Inhalte werden jedoch wie folgt behandelt:

- Chef ersetzt `<%= node[:attribute][:sub_attribute][:...]%>` durch den Wert des Attributs.

  So wird `StartServers <%= node[:apache][:prefork][:startservers] %>` in `httpd.conf` beispielsweise durch `StartServers 16` ersetzt.
- Mithilfe von `<%if-%>, <%else-%>, and <%end-%>` können Sie einen Wert anhand einer Bedingung auswählen.

  Im Beispiel wird abhängig von der Plattform ein anderer Pfad für `accept.lock` festgelegt.

> **Note**
>
> Sie sind nicht auf die Attribute in den Attributdateien Ihres Rezeptbuchs beschränkt. Sie können sämtliche Attribute im Knotenobjekt der Instance verwenden. Zum Beispiel, vom Chef-Tool Ohai generiert und ebenfalls im Knotenobjekt gespeichert. Weitere Informationen zu Attributen finden Sie unter Überschreiben der Attribute.

Weitere Informationen zu Vorlagen, einschließlich der Einbindung von Ruby-Code, finden Sie unter About Templates.

Rezepte

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Rezepte sind Ruby-Anwendungen zur Konfiguration des Systems. Sie installieren Pakete, erstellen Konfigurationsdateien aus Vorlagen, führen Shell-Befehle aus, erstellen Dateien und Verzeichnisse usw. Normalerweise lassen Sie AWS OpsWorks Stacks Rezepte automatisch ausführen, wenn ein Lebenszyklusereignis auf der Instance eintritt. Sie können sie aber auch jederzeit explizit ausführen, indem Sie den Stack-Befehl Execute Recipes verwenden. Weitere Informationen finden Sie unter About Recipes.

Ein Rezept besteht in der Regel hauptsächlich aus einer Reihe von Ressourcen, die jeweils für einen gewünschten Zustand eines Aspekts des Systems stehen. Jede Ressource enthält eine Reihe von Attributen, über die der gewünschte Zustand definiert und festgelegt wird, welche Aktionen dafür notwendig sind. Chef ordnet die Ressourcen einem geeigneten Anbieter zu, der die Aktion ausführt. Weitere Informationen finden Sie unter Resources and Providers Reference.

Mit `package`-Ressourcen können Sie Softwarepakete auf Linux-Instances verwalten. Im folgenden Beispiel wird das Apache-Paket installiert.

```
...
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 verwendet den korrekten Paketanbieter für die jeweilige Plattform. Ressourcenattributen wird oft nur ein Wert zugewiesen, Sie können Werte aber mithilfe logischer Ruby-Operatoren auch anhand bestimmter Bedingungen zuweisen. Im Beispiel wird der Operator `case` verwenden, der anhand von `node[:platform]` das Betriebssystem der Instance bestimmt und dem Attribut `package_name` den korrekten Wert zuweist. Sie können Attribute mithilfe der Standard-Chef-Knotensyntax in ein Rezept einfügen und Chef ersetzt diese durch die zugehörigen Werte. Sie können beliebige Attribute im Knotenobjekt verwenden und sind nicht auf die Attribute Ihres Rezeptbuchs beschränkt.

Nachdem der korrekte Paketname ermittelt wurde, endet das Codesegment mit einer `install`-Aktion, um das Paket zu installieren. Andere mögliche Aktionen dieser Ressource sind `upgrade` und `remove`. Weitere Informationen finden Sie unter package.

Oft kann es hilfreich sein, komplexe Installations- und Konfigurationsaufgaben aufzuteilen und die Unteraufgaben als eigene Rezepte zu implementieren. Mithilfe eines Hauptrezepts können Sie dann die einzelnen Unterrezepte zum richtigen Zeitpunkt ausführen. Das folgende Beispiel enthält die Codezeile nach dem vorherigen Beispiel:

```
include_recipe 'apache2::service'
```

Damit ein Rezept ein untergeordnetes Rezept ausführen kann, verwenden Sie das Schlüsselwort `include_recipe` gefolgt vom Rezeptnamen. Rezepte werden anhand der Standard-Chef-*`CookbookName`*`::`*`RecipeName`*-Syntax identifiziert, wobei bei *`RecipeName`* die Endung `.rb` weggelassen wird.

> **Note**
>
> Mit einer `include_recipe`-Anweisung führen Sie das Rezept an einer bestimmten Stelle im Hauptrezept aus. Tatsächlich ersetzt Chef jedoch jede `include_recipe`-Anweisung durch den Code des jeweiligen Rezepts, bevor das Hauptrezept ausgeführt wird.

Eine `directory`-Ressource steht für ein Verzeichnis, beispielsweise das Installationsverzeichnis für Paketdateien. Mit der folgenden `default.rb`-Ressource wird ein Linux-Protokollverzeichnis erstellt.

```
directory node[:apache][:log_dir] do
    mode 0755
    action :create
end
```

Das Protokollverzeichnis ist in einer Attributdatei des Rezeptbuchs definiert. Die Ressource legt den Verzeichnismodus auf 0755 fest und erstellt es mit der Aktion `create`. Weitere Informationen finden Sie unter directory. Diese Ressource kann auch auf Windows-Instances angewendet werden.

Die Ressource `execute` stellt Befehle wie Shell-Befehle oder Skripte dar. Im folgenden Beispiel werden module.load-Dateien erstellt.

```
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
```

Die Ressource bestimmt zunächst den CPU-Typ. `[:kernel][:machine]` ist ein weiteres automatisches Attribut, das von Chef für verschiedene Systemeigenschaften, in diesem Fall den CPU-Typ, erstellt wird. Danach wird der Befehl, ein Perl-Skript, festgelegt, das mit der Aktion `run` ausgeführt wird, um die module.load-Dateien zu erstellen. Weitere Informationen finden Sie unter execute.

Eine `template` Ressource stellt eine Datei dar — in der Regel eine Konfigurationsdatei —, die aus einer der Vorlagendateien des Kochbuches generiert werden soll. Im nachfolgenden Beispiel wird die Konfigurationsdatei `httpd.conf` aus der Vorlage `apache2.conf.erb` erstellt, die wir bereits in Vorlagen erläutert haben.

```
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
```

Über die Ressource wird der Name der generierten Datei sowie der Speicherort abhängig vom Betriebssystem der Instance festgelegt. Dann wird festgelegt, dass `apache2.conf.erb` als Vorlage zum Erstellen der Datei verwendet wird. Außerdem werden Eigentümer, Gruppe und Modus der Datei festgelegt. Dann wird `notify` ausgeführt, um die Ressource `service`, die für den Apache-Server steht, anzuweisen, den Server neu zu starten. Weitere Informationen finden Sie unter template.

Stack-Konfigurations- und Bereitstellungsattribute: Linux

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Thema werden die am häufigsten verwendeten Stack-Konfigurations- und Bereitstellungsattribute und deren Knotensyntax vorgestellt. Der Aufbau orientiert sich an der Struktur des Stack-Konfigurations-Namespace, der von Linux-Stacks verwendet wird. Dieselben Attributnamen werden teilweise für unterschiedliche Zwecke verwendet und tauchen in unterschiedlichen Namespaces auf. `id` kann sich beispielsweise auf eine Stack-ID, eine Layer-ID, eine App-ID usw. beziehen. Um diesen Attributwert verwenden zu können, benötigen Sie daher den vollständig qualifizierten Namen. Diese Daten lassen sich mit einem JSON-Objekt einfach darstellen. Beispiele finden Sie unter Attribute für die Stack-Konfiguration und -Bereitstellung.

> **Note**
>
> Auf Linux-Instances installiert AWS OpsWorks Stacks dieses JSON-Objekt auf jeder Instanz zusätzlich zum Hinzufügen der Daten zum Node-Objekt. Dieses können Sie mit dem Befehl `get_json` der Agenten-CLI abrufen.

Themen

- Andere Attribute auf oberster Ebene

opsworks-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Das `opsworks` Element — manchmal auch als `opsworks` Namespace bezeichnet — enthält eine Reihe von Attributen, die die grundlegende Stack-Konfiguration definieren.

> **Important**
>
> Es wird nicht empfohlen, die Attributwerte im `opsworks`-Namespace zu überschreiben, da dies dazu führen kann, dass die integrierten Rezepte nicht mehr funktionieren.

Themen

- applications
- Instance-Attribute
- Layer-Attribute
- rails_stack-Attribute
- Stack-Attribute
- Andere opsworks-Attribute auf oberster Ebene

applications

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir

empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Enthält eine Liste der eingebetteten Objekte, eines für jede App auf dem Stack. Jedes eingebettete Objekt enthält die folgenden Attribute, über die die Konfiguration der Anwendung festgelegt wird.

Note

Die allgemeine Knotensyntax dieser Attribute sieht wie folgt aus, wobei *i* den nullbasierten Listenindex der Instance angibt.

```
node["opsworks"]["applications"]["i"]["attribute_name"]
```

application_type

Der Anwendungstyp (Zeichenfolge). Die möglichen Werte lauten wie folgt:

- php: PHP-App
- `rails`: Eine Ruby on Rails-App
- `java`: Eine Java-App
- `nodejs`: Eine Node.js-App
- `web`: Eine statische HTML-Seite
- `other`: Alle anderen Anwendungstypen

```
node["opsworks"]["applications"]["i"]["application_type"]
```

Name

Der benutzerdefinierte Anzeigename, z. B. `"SimplePHP"` (Zeichenfolge)

```
node["opsworks"]["applications"]["i"]["name"]
```

slug_name

Ein Kurzname, bei dem es sich ausschließlich um einen Namen in Kleinbuchstaben handelt`"simplephp"`, der beispielsweise OpsWorks aus dem Namen der App (Zeichenfolge) generiert wird.

```
node["opsworks"]["applications"]["i"]["slug_name"]
```

Instance-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Das Attribut `instance` enthält eine Reihe von Attributen, über die die Konfiguration dieser Instance festgelegt wird.

Anwendung ansehen	availability_zone	backends
aws_instance_id	hostname	id
instance_type	ip	Ebenen
private_dns_name	private_ip	public_dns_name
Region		

Anwendung ansehen

Die Architektur der Instance, z. B. `"i386"` (Zeichenfolge)

```
node["opsworks"]["instance"]["architecture"]
```

availability_zone

Die Availability Zone der Instance, z. B. "us-west-2a" (Zeichenfolge)

```
node["opsworks"]["instance"]["availability_zone"]
```

backends

Die Anzahl der Back-End-Webprozesse (Zeichenfolge). Hierüber wird beispielsweise die Anzahl gleichzeitiger Verbindungen festgelegt, die HAProxy an ein Rails-Back-End weiterleitet. Der Wert hängt vom Arbeitsspeicher und der Anzahl der Kerne der Instance ab.

```
node["opsworks"]["instance"]["backends"]
```

aws_instance_id

Die EC2-Instance-ID (Zeichenfolge).

```
node["opsworks"]["instance"]["aws_instance_id"]
```

hostname

Der Host-Name, z. B. "php-app1" (Zeichenfolge)

```
node["opsworks"]["instance"]["hostname"]
```

id

Die Instanz-ID, bei der es sich um eine von AWS OpsWorks Stacks generierte GUID handelt, die die Instanz eindeutig identifiziert (Zeichenfolge).

```
node["opsworks"]["instance"]["id"]
```

instance_type

Der Instance-Typ, z. B. "c1.medium" (Zeichenfolge)

```
node["opsworks"]["instance"]["instance_type"]
```

ip

Die öffentliche IP-Adresse (Zeichenfolge).

```
node["opsworks"]["instance"]["ip"]
```

Ebenen

Alle Layer der Instance, erkennbar an ihren Kurznamen, z. B. `"lb"` oder `"db-master"` (Liste mit Zeichenfolgen)

```
node["opsworks"]["instance"]["layers"]
```

private_dns_name

Der private DNS-Name (Zeichenfolge).

```
node["opsworks"]["instance"]["private_dns_name"]
```

private_ip

Die private IP-Adresse (Zeichenfolge).

```
node["opsworks"]["instance"]["private_ip"]
```

public_dns_name

Der öffentliche DNS-Name (Zeichenfolge).

```
node["opsworks"]["instance"]["public_dns_name"]
```

Region

Die AWS-Region, z. B. `"us-west-2"` (Zeichenfolge)

```
node["opsworks"]["instance"]["region"]
```

Layer-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir

empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Das Attribut `layers` enthält eine Reihe von Layer-Attributen, eines für jeden Layer des Stacks, mit dem Kurznamen des Layers, z. B. `php-app`. Ein Stack kann maximal über eine Version der integrierten Layer mit den folgenden Kurznamen verfügen:

- `db-master`: MySQL-Schicht
- `java-app`: Java-App-Server-Ebene
- `lb`: HAProxy-Schicht
- `monitoring-master`: Ganglien-Schicht
- `memcached`: Memcached-Schicht
- `nodejs-app`: App-Server-Ebene von Node.js
- `php-app`: PHP-App-Server-Ebene
- `rails-app`: Rails-App-Server-Ebene
- `web`: Statische Webserver-Schicht

Ein Stack kann beliebig viele benutzerdefinierte Layer mit benutzerdefinierten Kurznamen enthalten.

Jedes Layer-Attribut enthält die folgenden Attribute:

- id
- -Instances
- Name

id

Die Layer-ID, bei der es sich um eine GUID handelt, die von der Ebene generiert wird OpsWorks und diese eindeutig identifiziert (Zeichenfolge).

```
node["opsworks"]["layers"]["layershortname"]["id"]
```

-Instances

Das `instances`-Element enthält eine Reihe von Instance-Attributen, eines für jede Online-Instances des Layers. Die Elemente sind nach dem Kurznamen der Instance benannt, z. B. `php-app1`.

> **Note**
>
> Das `instances`-Element enthält nur die Instances, die online sind, wenn die entsprechenden Stack-Konfigurations- und Bereitstellungsattribute erstellt werden.

Jedes Instance-Element enthält die folgenden Attribute:

availability_zone	aws_instance_id	backends
booted_at	created_at	elastic_ip
instance_type	ip	private_ip
public_dns_name	private_dns_name	Region
Status		

availability_zone

Die Availability Zone, z. B. `"us-west-2a"` (Zeichenfolge)

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["availability_zone"]
```

aws_instance_id

Die EC2-Instance-ID (Zeichenfolge).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["aws_instance_id"]
```

backends

Die Anzahl der Back-End-Webprozesse (Zahl). Hierüber wird beispielsweise die Anzahl gleichzeitiger Verbindungen festgelegt, die HAProxy an ein Rails-Back-End weiterleitet. Der Wert hängt vom Arbeitsspeicher und der Anzahl der Kerne der Instance ab.

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["backends"]
```

booted_at

Die Uhrzeit, zu der die EC2-Instance gestartet wurde, wobei das UTC-Format yyyy-mm-ddd THH:MM:SS+HH:MM (Zeichenfolge) verwendet wurde. Beispielsweise gibt der Wert `"2013-10-01T08:35:22+00:00"` den 10. Oktober 2013 um 8:35:22 Uhr ohne Zeitzonenabweichung an. Weitere Informationen finden Sie unter ISO 8601.

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["booted_at"]
```

created_at

Die Uhrzeit, zu der die EC2-Instance erstellt wurde, im UTC-Format thh:mm:ss+hh:mm (Zeichenfolge). yyyy-mm-ddd Beispielsweise gibt der Wert `"2013-10-01T08:35:22+00:00"` den 10. Oktober 2013 um 8:35:22 Uhr ohne Zeitzonenabweichung an. Weitere Informationen finden Sie unter ISO 8601.

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["created_at"]
```

elastic_ip

Die Elastic IP-Adresse, die auf null festgelegt ist, wenn die Instance über keine IP-Adresse verfügt (Zeichenfolge)

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["elastic_ip"]
```

instance_type

Der Instance-Typ, z. B. `"c1.medium"` (Zeichenfolge)

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["instance_type"]
```

ip

Die öffentliche IP-Adresse (Zeichenfolge).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["ip"]
```

private_ip

Die private IP-Adresse (Zeichenfolge).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["private_ip"]
```

public_dns_name

Der öffentliche DNS-Name (Zeichenfolge).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["public_dns_name"]
```

private_dns_name

Der private DNS-Name (Zeichenfolge).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["private_dns_name"]
```

Region

Die AWS-Region, z. B. `"us-west-2"` (Zeichenfolge)

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["region"]
```

Status

Der Status (Zeichenfolge) Die möglichen Werte lauten wie folgt:

- "requested"
- "booting"
- "running_setup"
- "online"
- "setup_failed"
- "start_failed"
- "terminating"
- "terminated"
- "stopped"
- "connection_lost"

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["status"]
```

Name

Der Layer-Name, mit dem der Layer in der Konsole angezeigt wird (Zeichenfolge). Er kann vom Benutzer festgelegt werden und muss nicht eindeutig sein.

```
node["opsworks"]["layers"]["layershortname"]["name"]
```

rails_stack-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Name

Legt den Rails-Stack fest, entweder "apache_passenger" oder "nginx_unicorn" (Zeichenfolge).

```
node["opsworks"]["rails_stack"]["name"]
```

recipe

Das zugehörige Rezept, abhängig davon, ob Sie Passenger oder Unicorn verwenden (Zeichenfolge):

- Unicorn: `"unicorn::rails"`
- Passenger: `"passenger_apache2::rails"`

```
node["opsworks"]["rails_stack"]["recipe"]
```

restart_command

Der Neustartbefehl, abhängig davon, ob Sie Passenger oder Unicorn verwenden (Zeichenfolge):

- Unicorn: `"../../shared/scripts/unicorn clean-restart"`
- Passenger: `"touch tmp/restart.txt"`

Service nicht zulässig

Der Service-Name, abhängig davon, ob Sie Passenger oder Unicorn verwenden (Zeichenfolge):

- Unicorn: `"unicorn"`
- Passenger: `"apache2"`

```
node["opsworks"]["rails_stack"]["service"]
```

Stack-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Über `stack`-Attribute legen Sie einige Aspekte der Stack-Konfiguration wie die Service-Layer-Konfiguration fest.

- elb-load-balancers
- id
- Name
- rds_instances
- vpc_id

elb-load-balancers

Enthält eine Liste von eingebetteten Objekten, eines für jeden Elastic Load Balancing-Load Balancer im Stack. Jedes eingebettete Objekt enthält die folgenden Attribute, die die Load Balancer-Konfiguration festlegen.

> **Note**
>
> Die allgemeine Knotensyntax dieser Attribute sieht wie folgt aus, wobei *i* den nullbasierten Listenindex der Instance angibt.
>
> ```
> node["opsworks"]["stack"]["elb-load-balancers"]["i"]["attribute_name"]
> ```

dns_name

Der DNS-Name des Load Balancers (Zeichenfolge).

```
node["opsworks"]["stack"]["elb-load-balancers"]["i"]["dns_name"]
```

Name

Der Load Balancer-Name (Zeichenfolge).

```
node["opsworks"]["stack"]["elb-load-balancers"]["i"]["name"]
```

layer_id

Die ID des Layers, mit dem der Load Balancer verknüpft ist (Zeichenfolge)

```
node["opsworks"]["stack"]["elb-load-balancers"]["i"]["layer_id"]
```

id

Die Stack-ID (Zeichenfolge)

```
node["opsworks"]["stack"]["id"]
```

Name

Der Stack-Name (Zeichenfolge)

```
node["opsworks"]["stack"]["name"]
```

rds_instances

Enthält eine Liste von eingebetteten Objekten, eines für jede Amazon RDS-Instance, die bei dem Stack registriert ist. Jedes eingebettete Objekt enthält eine Reihe von Attributen zur Festlegung der Instance-Konfiguration. Sie können diese Werte beim Erstellen der Instance auf der Amazon RDS-Konsole oder API festlegen. Sie können auch die Amazon RDS-Konsole oder API verwenden, um einige Einstellungen zu bearbeiten, nachdem die Instance erstellt wurde. Weitere Informationen finden Sie in der Dokumentation zu Amazon RDS.

> **Note**
>
> Die allgemeine Knotensyntax dieser Attribute sieht wie folgt aus, wobei *i* den nullbasierten Listenindex der Instance angibt.
>
> ```
> node["opsworks"]["stack"]["rds_instances"]["i"]["attribute_name"]
> ```

Wenn Ihr Stack mehrere Amazon RDS-Instances enthält, finden Sie im Folgenden ein Beispiel für die Verwendung einer bestimmten Instance in einem Rezept.

```
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	Region
status_infos	vpc_security_groups	

address

Die Instance-URL, z. B. `opsinstance.ccdvt3hwog1a.us-west-2.rds.amazonaws.com` (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["address"]
```

allocated_storage

Der zugewiesene Speicher in GB (Zahl)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["allocated_storage"]
```

arn

Der Instance-ARN (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["arn"]
```

auto_minor_version_upgrade

Legt fest, ob geringfügige Versions-Upgrades automatisch angewendet werden (boolescher Wert)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["auto_minor_version_upgrade"]
```

availability_zone

Die Availability Zone der Instance, z. B. `us-west-2a` (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["availability_zone"]
```

backup_retention_period

Der Aufbewahrungszeitraum für Backups in Tagen (Zahl)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["backup_retention_period"]
```

db_instance_class

Die DB-Instance-Klasse, z. B. `db.m1.small` (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_instance_class"]
```

db_instance_identifier

Die benutzerdefinierte DB-Instance-Kennung (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_instance_identifier"]
```

db_instance_status

Der Status der Instance (Zeichenfolge). Weitere Informationen finden Sie unter DB-Instance.

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_instance_status"]
```

db_name

Der benutzerdefinierten DB-Name (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_name"]
```

db_parameter_groups

Die DB-Parametergruppen der Instance mit einer Liste von eingebetteten Objekten, eines für jede Parametergruppe. Weitere Informationen finden Sie unter Arbeiten mit DB-Parametergruppen. Jedes Objekt enthält die folgenden Attribute:

db_parameter_group_name

Der Gruppenname (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_parameter_groups"][j"]
["db_parameter_group_name"]
```

parameter_apply_status

Der tatsächliche Status (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_parameter_groups"][j"]
["parameter_apply_status"]
```

db_security_groups

Die Sicherheitsgruppen der Instance-Datenbank mit einer Liste der eingebetteten Objekte, eines für jede Sicherheitsgruppe. Weitere Informationen finden Sie unter Arbeiten mit DB-Sicherheitsgruppen. Jedes Objekt enthält die folgenden Attribute:

db_security_group_name

Der Name der Sicherheitsgruppe (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_security_groups"][j"]
["db_security_group_name"]
```

Status

Der Status (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_security_groups"][j"]
["status"]
```

db_user

Der benutzerdefinierte Master-Benutzername (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_user"]
```

engine

Die Datenbank-Engine, z. B. `mysql(5.6.13)` (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["engine"]
```

instance_create_time

Der Erstellungszeitpunkt der Datenbank, z. B. `2014-04-15T16:13:34Z` (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["instance_create_time"]
```

license_model

Das Lizenzmodell der Instance, z. B. `general-public-license` (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["license_model"]
```

multi_az

Legt fest, ob Multi-AZ-Bereitstellung aktiviert ist (boolescher Wert)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["multi_az"]
```

option_group_memberships

Die Mitgliedschaften der Instance in Optionsgruppen mit einer Liste von eingebetteten Objekten, eines für jede Optionsgruppe. Weitere Informationen finden Sie unter Arbeiten mit Optionsgruppen. Jedes Objekt enthält die folgenden Attribute:

option_group_name

Der Gruppenname (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["option_group_memberships"]
[j"]["option_group_name"]
```

Status

Der Gruppenstatus (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["option_group_memberships"]
[j"]["status"]
```

port

Der Serverport der Datenbank (Zahl)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["port"]
```

preferred_backup_window

Das bevorzugte Zeitfenster für die tägliche Datensicherung, z. B. `06:26-06:56` (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["preferred_backup_window"]
```

preferred_maintenance_window

Das bevorzugte Zeitfenster für wöchentliche Wartungsarbeiten, z. B. `thu:07:13-thu:07:43` (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["preferred_maintenance_window"]
```

publicly_accessible

Legt fest, ob die Datenbank öffentlich zugreifbar ist (boolescher Wert).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["publicly_accessible"]
```

read_replica_db_instance_identifiers

Eine Liste der Read Replica-Instance-Kennungen (Liste mit Zeichenfolgen). Weitere Informationen finden Sie unter Arbeiten mit Read Replicas.

```
node["opsworks"]["stack"]["rds_instances"]["i"]
["read_replica_db_instance_identifiers"]
```

Region

Die AWS-Region, z. B. `us-west-2` (Zeichenfolge)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["region"]
```

status_infos

Eine Liste mit Statusinformationen (Liste mit Zeichenfolgen)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["status_infos"]
```

vpc_security_groups

Eine Liste der VPC-Sicherheitsgruppen (Liste mit Zeichenfolgen)

```
node["opsworks"]["stack"]["rds_instances"]["i"]["vpc_security_groups"]
```

vpc_id

Die VPC-ID (Zeichenfolge). Dieser Wert ist `null`, wenn die Instance nicht in einer VPC ausgeführt wird.

```
node["opsworks"]["stack"]["vpc_id"]
```

Andere opsworks-Attribute auf oberster Ebene

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Abschnitt werden die `opsworks`-Attribute ohne untergeordnete Attribute vorgestellt.

Aktivität

Die Aktivität, die den Attributen zugeordnet ist, z. B. `deploy` (Zeichenfolge)

```
node["opsworks"]["activity"]
```

agent_version

Die Version des OpsWorks Agenten der Instanz (Zeichenfolge).

```
node["opsworks"]["agent_version"]
```

deploy_chef_provider

Der Chef-Bereitstellungsanbieter, der sich auf die Verzeichnisstruktur von bereitgestellten Apps auswirkt (Zeichenfolge). Sie können dieses Attribut auf eines der folgenden Werte setzen:

- `Branch`
- `Revision`
- `Timestamped` (Standardwert)

```
node["opsworks"]["deploy_chef_provider"]
```

ruby_stack

Der Ruby-Stack (Zeichenfolge). Die Standardeinstellung ist die Enterprise-Version (`ruby_enterprise`). Wenn Sie die MRI-Version verwenden möchten, wählen Sie die Option `ruby` aus.

```
node["opsworks"]["ruby_stack"]
```

ruby_version

Die Ruby-Version, die von Anwendungen verwendet wird (Zeichenfolge). Mithilfe dieses Attributs können Sie die Haupt- und Nebenversion festlegen. Verwenden Sie das korrekte ["ruby"]-Attribut, um die Patch-Version festzulegen. Weitere Informationen zum Festlegen von Versionen, einschließlich Beispielen, finden Sie unter Ruby-Versionen. Ausführliche Informationen darüber, wie AWS OpsWorks Stacks die Ruby-Version bestimmt, finden Sie in der integrierten Attributdatei ruby.rb.

```
node["opsworks"]["ruby_version"]
```

run_cookbook_tests

Ob Sie minitest-chef-handlerTests für Ihre Chef 11.4-Kochbücher (Boolean) ausführen möchten.

```
node["opsworks"]["run_cookbook_tests"]
```

sent_at

Gibt an, wann dieser Befehl an die Instance gesendet wurde (Zahl).

```
node["opsworks"]["sent_at"]
```

Bereitstellung

Wenn diese Attribute einer Bereitstellungsaktivität zugeordnet sind, ist `deployment` die Bereitstellungs-ID, eine von AWS OpsWorks Stacks generierte GUID, über die die Bereitstellung eindeutig identifiziert wird (Zeichenfolge). Andernfalls ist dieses Attribut gleich null.

```
node["opsworks"]["deployment"]
```

opsworks_custom_cookbooks-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Enthält Attribute, über die die benutzerdefinierten Rezeptbücher des Stacks festgelegt werden.

aktiviert

Gibt an, ob benutzerdefinierte Rezeptbücher aktiviert sind (Boolescher Wert).

```
node["opsworks_custom_cookbooks"]["enabled"]
```

recipes

Eine Liste der Rezepte, einschließlich benutzerdefinierter Rezepte, die mit diesem Befehl ausgeführt werden, im Format *cookbookname*::*recipename* (Liste mit Zeichenfolgen)

```
node["opsworks_custom_cookbooks"]["recipes"]
```

Abhängigkeitsattribute

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Enthält mehrere Attribute, die zum `update_dependencies`Stack-BefehlAusführen von Stack-Befehlen gehören.

gem_binary

Der Speicherort der Gems-Binärdatei (Zeichenfolge)

upgrade_debs

Legt fest, ob Debs-Paket-Upgrades ausgeführt werden (boolescher Wert).

update_debs

Legt fest, ob Debs-Pakete aktualisiert werden (boolescher Wert).

ganglia-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Enthält ein web-Attribut, das wiederum mehrere Attribute enthält, über die der Zugriff auf die Statistik-Webseite des Ganglia-Servers geregelt wird:

password

Das für den Zugriff auf die Statistikseite erforderliche Passwort (Zeichenfolge)

```
node["ganglia"]["web"]["password"]
```

URL

Der URL-Pfad der Statistikseite, z. B. `"/ganglia"` (Zeichenfolge). Die vollständige URL lautet `http://`*`DNSNameURLPath`*, wobei *`DNSName`* der DNS-Name der zugehörigen Instance ist.

```
node["ganglia"]["web"]["url"]
```

user

Der Benutzername, der für den Zugriff auf die Statistikseite erforderlich ist (Zeichenfolge)

```
node["ganglia"]["web"]["user"]
```

mysql-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Enthält eine Reihe von Attributen, über die die Konfiguration des MySQL-Datenbankservers festgelegt wird.

clients

Eine Liste der Client-IP-Adressen (Liste mit Zeichenfolgen)

```
node["mysql"]["clients"]
```

server_root_password

Das Root-Passwort (Zeichenfolge)

```
node["mysql"]["server_root_password"]
```

passenger-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Enthält eine Reihe von Attributen, über die die Phusion Passenger-Konfiguration festgelegt wird.

gem_bin

Der Speicherort der RubyGems Binärdateien, z. B. "`/usr/local/bin/gem`" (Zeichenfolge).

```
node["passenger"]["gem_bin"]
```

max_pool_size

Die maximale Pool-Größe (Zahl)

```
node["passenger"]["max_pool_size"]
```

ruby_bin

Der Speicherort der Ruby-Binärdateien, z. B. "`/usr/local/bin/ruby`"

```
node["passenger"]["ruby_bin"]
```

version

Die Passenger-Version (Zeichenfolge)

```
node["passenger"]["version"]
```

opsworks_bundler-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Enthält Elemente zur Konfiguration des Bundler-Supports.

manage_package

Legt fest, ob Bundler installiert und verwaltet werden soll (boolescher Wert).

```
node["opsworks_bundler"]["manage_package"]
```

version

Die Bundler-Version (Zeichenfolge)

```
node["opsworks_bundler"]["version"]
```

Bereitstellungsattribute

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn diese Attribute einem Bereitstellungsereignis oder einem Stack-Befehl "Execute Recipes" zugeordnet werden, enthält das Attribut `deploy` ein Attribut für jede bereitgestellte App mit dem Kurznamen der App. Jedes App-Attribut enthält die folgenden Attribute:

Anwendung	application_type	auto_bundle_on_deploy
Datenbank	deploy_to	domains
document_root	environment_variables	Gruppe
keep_releases	memcached	migrate
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
user		

Anwendung

Der Slug-Name der App, z. B. `"simplephp"` (Zeichenfolge)

```
node["deploy"]["appshortname"]["application"]
```

application_type

Der App-Typ (Zeichenfolge). Die möglichen Werte lauten wie folgt:

- `java`: Eine Java-App
- `nodejs`: Eine Node.js-App
- `php`: Ein PHP-App
- `rails`: Eine Ruby on Rails-App
- `web`: Eine statische HTML-Seite
- `other`: Alle anderen Anwendungstypen

```
node["deploy"]["appshortname"]["application_type"]
```

auto_bundle_on_deploy

Legt für Rails-Anwendungen fest, ob Bundler während der Bereitstellung ausgeführt wird (boolescher Wert).

```
node["deploy"]["appshortname"]["auto_bundle_on_deploy"]
```

Datenbank

Enthält die für die Verbindung zur App-Datenbank erforderlichen Informationen. Wenn der App eine Datenbankschicht angehängt ist, weist AWS OpsWorks Stacks diesen Attributen automatisch die entsprechenden Werte zu.

adapter

Der Datenbank Adapter, z. B. `mysql` (Zeichenfolge)

```
node["deploy"]["appshortname"]["database"]["adapter"]
```

Datenbank

Der Datenbankname, in der Regel der Slug-Name der App, z. B. `"simplephp"` (Zeichenfolge)

```
node["deploy"]["appshortname"]["database"]["database"]
```

data_source_provider

Die Datenquelle: `mysql` oder `rds` (Zeichenfolge)

```
node["deploy"]["appshortname"]["database"]["data_source_provider"]
```

Host

Die Host-IP-Adresse der Datenbank (Zeichenfolge)

```
node["deploy"]["appshortname"]["database"]["host"]
```

password

Das Datenbankpasswort (Zeichenfolge)

```
node["deploy"]["appshortname"]["database"]["password"]
```

port

Der Datenbank-Port (Zahl)

```
node["deploy"]["appshortname"]["database"]["port"]
```

reconnect

Legt für Rails-Anwendungen fest, ob die Anwendung nach einem Verbindungsabbruch eine neue Verbindung herstellt (boolescher Wert).

```
node["deploy"]["appshortname"]["database"]["reconnect"]
```

username

Der Benutzername (Zeichenfolge).

```
node["deploy"]["appshortname"]["database"]["username"]
```

deploy_to

Legt fest, wo die App bereitgestellt wird, z. B. `"/srv/www/simplephp"` (Zeichenfolge).

```
node["deploy"]["appshortname"]["deploy_to"]
```

domains

Eine Liste der App-Domänen (Liste aus Zeichenfolgen)

```
node["deploy"]["appshortname"]["domains"]
```

document_root

Das Dokumenten-Stammverzeichnis, wenn Sie vom Standardstammverzeichnis abweichen, oder null, wenn Sie das Standardverzeichnis verwenden (Zeichenfolge)

```
node["deploy"]["appshortname"]["document_root"]
```

environment_variables

Eine Sammlung von bis zu zwanzig Attributen, die die benutzerdefinierten Umgebungsvariablen für die App festlegen. Weitere Informationen zur Definition von Umgebungsvariablen für eine App finden Sie unter Hinzufügen von Apps. Jeder Attributname entspricht dem Namen einer Umgebungsvariable und der entsprechende Wert entspricht dem Variablenwert. Daher können Sie mit der folgenden Syntax auf einen bestimmten Wert verweisen.

```
node["deploy"]["appshortname"]["environment_variables"]["variable_name"]
```

Gruppe

Die App-Gruppe (Zeichenfolge)

```
node["deploy"]["appshortname"]["group"]
```

keep_releases

Die Anzahl der App-Bereitstellungen, die AWS OpsWorks Stacks speichern wird (Anzahl). Dieses Attribut bestimmt, wie oft Sie ein Rollback für eine Anwendung ausführen können. Standardmäßig ist dies der globale Wert, deploy_keep_releases , mit einem Standardwert von 5. Sie können `keep_releases` überschreiben, um die Anzahl der Bereitstellungen für eine bestimmte Anwendung anzupassen.

```
node["deploy"]["appshortname"]["keep_releases"]
```

memcached

Enthält zwei Attribute, über die die Memcached-Konfiguration festgelegt wird.

Host

Die IP-Adresse (Zeichenfolge) der Memcached-Serverinstanz.

```
node["deploy"]["appshortname"]["memcached"]["host"]
```

port

Der Port des Memcached-Servers (Zahl)

```
node["deploy"]["appshortname"]["memcached"]["port"]
```

migrate

Legt für Rails-Anwendungen fest, ob Migrationen ausgeführt werden (boolescher Wert).

```
node["deploy"]["appshortname"]["migrate"]
```

mounted_at

Der Mount-Punkt der App, falls Sie einen abweichenden Mount-Punkt festlegen, oder null, wenn Sie den Standardpunkt verwenden (Zeichenfolge)

```
node["deploy"]["appshortname"]["mounted_at"]
```

purge_before_symlink

Legt für Rails-Apps eine Reihe von Pfaden fest, die vor dem Erstellen von symbolischen Links bereinigt werden (Liste aus Zeichenfolgen).

```
node["deploy"]["appshortname"]["purge_before_symlink"]
```

rails_env

Für Rails App Server-Instanzen die Rails-Umgebung, z. B. `"production"` (string).

```
node["deploy"]["appshortname"]["rails_env"]
```

restart_command

Ein Befehl, der beim Neustart der App ausgeführt wird, z. B. `"echo 'restarting app'"`

```
node["deploy"]["appshortname"]["restart_command"]
```

scm

Enthält eine Reihe von Attributen, die die Informationen angeben, die zur Bereitstellung der App aus ihrem Quellcodeverwaltungs-Repository OpsWorks verwendet werden. Die Attribute sind abhängig vom Repository-Typ.

password

Das Passwort für private Repositorys und null für öffentliche Repositorys (Zeichenfolge). Für private Amazon S3 S3-Buckets ist das Attribut auf den geheimen Schlüssel gesetzt.

```
node["deploy"]["appshortname"]["scm"]["password"]
```

Repository

Die Repository-URL, z. B. `"git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git"` (Zeichenfolge)

```
node["deploy"]["appshortname"]["scm"]["repository"]
```

Änderung

Falls das Repository über mehrere Branches verfügt, gibt das Attribut den Branch oder die Version der App an, z. B. `"version1"` (Zeichenfolge). Andernfalls ist es auf null festgelegt.

```
node["deploy"]["appshortname"]["scm"]["revision"]
```

scm_type

Der Repository-Typ (Zeichenfolge). Die möglichen Werte lauten wie folgt:

- `"git"`: Ein Git-Repository
- `"svn"`: Ein Subversion-Repository

- `"s3"`: Ein Amazon S3 S3-Bucket
- `"archive"`: Ein HTTP-Archiv
- `"other"`: Ein anderer Repository-Typ

```
node["deploy"]["appshortname"]["scm"]["scm_type"]
```

ssh_key

Ein SSH-Bereitstellungsschlüssel für den Zugriff auf private Git-Repositorys und null für öffentliche Repositorys (Zeichenfolge)

```
node["deploy"]["appshortname"]["scm"]["ssh_key"]
```

user

Der Benutzername für private Repositorys und null für öffentliche Repositorys (Zeichenfolge). Für private Amazon S3 S3-Buckets ist das Attribut auf den Zugriffsschlüssel gesetzt.

```
node["deploy"]["appshortname"]["scm"]["user"]
```

ssl_certificate

Das SSL-Zertifikat der App, falls Sie SSL-Unterstützung aktiviert haben, andernfalls null (Zeichenfolge)

```
node["deploy"]["appshortname"]["ssl_certificate"]
```

ssl_certificate_ca

Sofern SSL aktiviert ist, legt dieses Attribut den Zertifizierungsstellenschlüssel des Zwischenzertifikats oder die Clientauthentifizierung fest (Zeichenfolge).

```
node["deploy"]["appshortname"]["ssl_certificate_ca"]
```

ssl_certificate_key

Der private SSL-Schlüssel der App, falls Sie SSL-Unterstützung aktiviert haben, andernfalls null (Zeichenfolge)

```
node["deploy"]["appshortname"]["ssl_certificate_key"]
```

ssl_support

Legt fest, ob SSL unterstützt wird (boolescher Wert).

```
node["deploy"]["appshortname"]["ssl_support"]
```

Stack

Enthält ein boolesches Attribut, `needs_reload`, über das festgelegt wird, ob der Anwendungsserver während der Bereitstellung erneut geladen wird.

```
node["deploy"]["appshortname"]["stack"]["needs_reload"]
```

symlink_before_migrate

Enthält für Rails-Apps symbolische Links, die vor dem Ausführen von Migrationen erstellt werden, z. B. "*link*":"*target*"-Paare.

```
node["deploy"]["appshortname"]["symlink_before_migrate"]
```

symlinks

Enthält die symbolischen Links der Bereitstellung als "*link*":"*target*"-Paare.

```
node["deploy"]["appshortname"]["symlinks"]
```

user

Der App-Benutzer (Zeichenfolge)

```
node["deploy"]["appshortname"]["user"]
```

Andere Attribute auf oberster Ebene

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Dieser Abschnitt enthält Stack-Konfigurationsattribute auf oberster Ebene ohne untergeordnete Attribute.

rails-Attribute

Enthält ein Attribut max_pool_size, über das die maximale Pool-Größe des Servers festgelegt wird (Zahl). Der Attributwert wird von AWS OpsWorks Stacks festgelegt und hängt vom Instanztyp ab. Sie können ihn jedoch überschreiben, indem Sie eine benutzerdefinierte JSON-Datei oder eine benutzerdefinierte Attributdatei verwenden.

```
node["rails"]["max_pool_size"]
```

recipes-Attribute

Eine Liste der integrierten Rezepte, die durch diese Aktivität ausgeführt werden, im Format "*cookbookname*::*recipename*" (Liste aus Zeichenfolgen)

```
node["recipes"]
```

opsworks_rubygems-Attribute

Enthält ein Versionselement, das die RubyGems Version (Zeichenfolge) angibt.

```
node["opsworks_rubygems"]["version"]
```

languages-Attribute

Enthält ein Attribut für jede installierte Sprache mit dem Namen der Sprache, z. B. ruby. Das Attribut ist ein Objekt, das ein Attribut enthält, z. B. ruby_bin, das wiederum das Installationsverzeichnis festlegt, z. B. "/usr/bin/ruby" (Zeichenfolge).

ssh_users-Attribute

Enthält eine Reihe von Attributen, von denen jedes einen der Benutzer beschreibt, denen SSH-Berechtigungen erteilt wurden. Jedes Attribut ist mit der Unix-ID eines Benutzers benannt. AWS OpsWorks Stacks generiert eine eindeutige ID für jeden Benutzer im Bereich von 2000 bis 4000,

z. B."2001", und erstellt für jede Instanz einen Benutzer mit dieser ID. Da der Bereich 2000-4000 AWS OpsWorks reserviert ist, können Benutzer, die Sie außerhalb von erstellen AWS OpsWorks (z. B. mithilfe von Kochbuchrezepten oder durch Import von Benutzern AWS OpsWorks aus IAM), UIDs haben, die von Stacks für einen anderen Benutzer überschrieben werden. AWS OpsWorks Es hat sich bewährt, Benutzer zu erstellen und ihren Zugriff in der Stacks-Konsole zu verwalten. AWS OpsWorks Wenn Sie Benutzer außerhalb von AWS OpsWorks Stacks erstellen, verwenden Sie *UnixID-Werte* über 4000.

Jedes Attribut enthält die folgenden Attribute:

email

Die E-Mail-Adresse des -Benutzers (Zeichenfolge)

```
node["ssh_users"]["UnixID"]["email"]
```

public_key

Der öffentliche SSH-Schlüssel des -Benutzers (Zeichenfolge)

```
node["ssh_users"]["UnixID"]["public_key"]
```

sudoer

Legt fest, ob der -Benutzer über sudo-Berechtigungen verfügt (boolescher Wert).

```
node["ssh_users"]["UnixID"]["sudoer"]
```

Name

Der -Benutzername (Zeichenfolge)

```
node["ssh_users"]["UnixID"]["name"]
```

Integrierte Rezeptbuchattribute

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir

empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Die meisten dieser Attribute stehen nur für Linux-Stacks zur Verfügung.

Die meisten integrierten Rezepte verfügen über ein oder mehrere Attributdateien, die verschiedene Einstellungen definieren. Sie können auf diese Einstellungen in Ihren benutzerdefinierten Rezepten zugreifen und das benutzerdefinierte JSON-Objekt verwenden, um sie zu überschreiben. In der Regel müssen Sie auf Attribute zugreifen oder diese überschreiben, die die Konfiguration der verschiedenen Servertechnologien steuern, die von AWS OpsWorks Stacks unterstützt werden. Dieser Abschnitt bietet eine Übersicht über diese Attribute. Die vollständigen Attributdateien und die zugehörigen Rezepte und Vorlagen sind unter https://github.com/aws/opsworks-cookbooks.git verfügbar.

> **Note**
>
> Alle integrierten Rezeptattribute sind vom Typ `default`.

Themen

- unicorn-Attribute

apache2-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Attribute stehen nur für Linux-Stacks zur Verfügung.

Die apache2-Attribute legen die Konfiguration des Apache HTTP-Servers fest. Weitere Informationen finden Sie unter Apache-Core-Funktionen. Weitere Informationen zum Überschreiben integrierter Attribute, um benutzerdefinierte Werte anzugeben, finden Sie unter Überschreiben der Attribute.

Binary	contact	deflate_types
dir	document_root	Gruppe
hide_info_headers	icondir	init_script
keepalive	keepaliverequests	keepalivetimeout
lib_dir	libexecdir	listen_ports
log_dir	logrotate-Attribute	pid_file
prefork-Attribute	serversignature	servertokens
timeout	traceenable	user
version	worker-Attribute	

Binary

Der Speicherort der Apache-Binärdatei (Zeichenfolge). Der Standardwert ist `'/usr/sbin/httpd'`.

```
node[:apache][:binary]
```

contact

Eine E-Mail-Kontaktadresse (Zeichenfolge). Der Standardwert ist eine Musteradresse, `'ops@example.com'`.

```
node[:apache][:contact]
```

deflate_types

Weist `mod_deflate` an, die Komprimierung für die angegebenen MIME-Typen zu aktivieren, wenn diese vom Browser unterstützt werden (Liste mit Zeichenfolgen). Der Standardwert lautet wie folgt:

```
['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

Die Komprimierung kann Sicherheitsrisiken bergen. Um die Komprimierung vollständig zu deaktivieren, legen Sie dieses Attribut wie folgt fest:

```
node[:apache][:deflate_types] = []
```

```
node[:apache][:deflate_types]
```

dir

Das Stammverzeichnis des Servers (Zeichenfolge). Die Standardwerte lauten wie folgt:

- Amazon Linux und Red Hat Enterprise Linux (RHEL): `'/etc/httpd'`
- Ubuntu: `'/etc/apache2'`

```
node[:apache][:dir]
```

document_root

Das Dokumenten-Stammverzeichnis (Zeichenfolge). Die Standardwerte lauten wie folgt:

- Amazon Linux und RHEL: `'/var/www/html'`
- Ubuntu: `'/var/www'`

```
node[:apache][:document_root]
```

Gruppe

Der Gruppenname (Zeichenfolge) Die Standardwerte lauten wie folgt:

- Amazon Linux und RHEL: `'apache'`
- Ubuntu: `'www-data'`

```
node[:apache][:group]
```

hide_info_headers

Gibt an, ob Versions- und Modulinformationen in HTTP-Headern ausgelassen werden (`'true'`/`'false'`) (Zeichenfolge). Der Standardwert ist `'true'`.

```
node[:apache][:hide_info_headers]
```

icondir

Das Symbolverzeichnis (Zeichenfolge). Der Standardwert lautet wie folgt:

- Amazon Linux und RHEL: `'/var/www/icons/'`
- Ubuntu: `'/usr/share/apache2/icons'`

```
node[:apache][:icondir]
```

init_script

Das Initialisierungsskript (Zeichenfolge). Die Standardwerte lauten wie folgt:

- Amazon Linux und RHEL: `'/etc/init.d/httpd'`
- Ubuntu: `'/etc/init.d/apache2'`

```
node[:apache][:init_script]
```

keepalive

Gibt an, ob Keepalive-Verbindungen aktiviert werden sollen (Zeichenfolge). Die möglichen Werte sind `'On'` und `'Off'` (Zeichenfolge). Der Standardwert ist `'Off'`.

```
node[:apache][:keepalive]
```

keepaliverequests

Die maximale Anzahl von Keepalive-Anforderungen, die Apache gleichzeitig verarbeitet (Zahl). Der Standardwert ist `100`.

```
node[:apache][:keepaliverequests]
```

keepalivetimeout

Die Zeit, die Apache vor dem Schließen der Verbindung auf eine Anforderung wartet (Zahl). Der Standardwert ist `3`.

```
node[:apache][:keepalivetimeout]
```

lib_dir

Das Verzeichnis, in dem sich die Objektcode-Bibliotheken befinden (Zeichenfolge). Die Standardwerte lauten wie folgt:

- Amazon Linux (x86): `'/usr/lib/httpd'`
- Amazon Linux (x64) und RHEL: `'/usr/lib64/httpd'`
- Ubuntu: `'/usr/lib/apache2'`

```
node[:apache][:lib_dir]
```

libexecdir

Das Verzeichnis, in dem sich die ausführbaren Dateien befinden (Zeichenfolge). Die Standardwerte lauten wie folgt:

- Amazon Linux (x86): `'/usr/lib/httpd/modules'`
- Amazon Linux (x64) und RHEL: `'/usr/lib64/httpd/modules'`
- Ubuntu: `'/usr/lib/apache2/modules'`

```
node[:apache][:libexecdir]
```

listen_ports

Eine Liste der Ports, die der Server überwacht (Liste mit Zeichenfolgen). Der Standardwert ist `[ '80','443' ]`.

```
node[:apache][:listen_ports]
```

log_dir

Das Protokollverzeichnis (Zeichenfolge). Die Standardwerte lauten wie folgt:

- Amazon Linux und RHEL: `'/var/log/httpd'`
- Ubuntu: `'/var/log/apache2'`

```
node[:apache][:log_dir]
```

logrotate-Attribute

Diese Attribute geben an, wie die Protokolldateien rotieren sollen.

delaycompress

Gibt an, ob die Komprimierung einer geschlossenen Protokolldatei bis zum Start des nächsten Rotationszyklus verzögert werden soll (`'true'`/`'false'`) (Zeichenfolge). Der Standardwert ist `'true'`.

```
node[:apache][:logrotate][:delaycompress]
```

Gruppe

Die Gruppe der Protokolldateien (Zeichenfolge). Der Standardwert ist `'adm'`.

```
node[:apache][:logrotate][:group]
```

mode

Der Modus der Protokolldateien (Zeichenfolge). Der Standardwert ist `'640'`.

```
node[:apache][:logrotate][:mode]
```

owner

Der Eigentümer der Protokolldateien (Zeichenfolge). Der Standardwert ist `'root'`.

```
node[:apache][:logrotate][:owner]
```

rotate

Die Anzahl der Rotationszyklen, bevor eine geschlossene Protokolldatei gelöscht wird (Zeichenfolge). Der Standardwert ist `'30'`.

```
node[:apache][:logrotate][:rotate]
```

schedule

Der Rotationsplan (Zeichenfolge). Die möglichen Werte lauten wie folgt:

- `'daily'`
- `'weekly'`
- `'monthly'`

Der Standardwert ist `'daily'`.

```
node[:apache][:logrotate][:schedule]
```

pid_file

Die Datei mit der Prozess-ID des Daemons (Zeichenfolge). Die Standardwerte lauten wie folgt:

- Amazon Linux und RHEL: `'/var/run/httpd/httpd.pid'`

- Ubuntu: `'/var/run/apache2.pid'`

```
node[:apache][:pid_file]
```

prefork-Attribute

Diese Attribute geben die Pre-Forking-Konfiguration an.

maxclients

Die maximale Anzahl von gleichzeitigen Anforderungen, die verarbeitet werden (Zahl). Der Standardwert ist `400`.

> **Note**
>
> Verwenden Sie dieses Attribut nur für Instances, auf denen Amazon Linux oder RHEL ausgeführt wird. Wenn Ihre Instances Ubuntu 14.04 LTS ausführen, verwenden Sie maxrequestworkers.

```
node[:apache][:prefork][:maxclients]
```

maxrequestsperchild

Die maximale Anzahl von Anforderungen, die ein untergeordneter Serverprozess verarbeiten kann (Zahl). Der Standardwert ist `10000`.

```
node[:apache][:prefork][:maxrequestsperchild]
```

maxrequestworkers

Die maximale Anzahl von gleichzeitigen Anforderungen, die verarbeitet werden (Zahl). Der Standardwert ist `400`.

> **Note**
>
> Verwenden Sie dieses Attribut nur für Instances, die Ubuntu 14.04 LTS ausführen. Wenn auf Ihren Instances Amazon Linux oder RHEL ausgeführt wird, verwenden Siemaxclients .

```
node[:apache][:prefork][:maxrequestworkers]
```

maxspareservers

Die maximale Anzahl von nicht aktiven untergeordneten Serverprozessen (Zahl). Der Standardwert ist `32`.

```
node[:apache][:prefork][:maxspareservers]
```

minspareservers

Die Mindestanzahl von nicht aktiven untergeordneten Serverprozessen (Zahl). Der Standardwert ist `16`.

```
node[:apache][:prefork][:minspareservers]
```

serverlimit

Die maximale Anzahl von Prozessen, die konfiguriert werden können (Zahl). Der Standardwert ist `400`.

```
node[:apache][:prefork][:serverlimit]
```

startservers

Die Anzahl von untergeordneten Serverprozessen, die beim Starten erstellt werden können (Zahl). Der Standardwert ist `16`.

```
node[:apache][:prefork][:startservers]
```

serversignature

Gibt an, ob und wie eine nachgestellte Fußzeile für vom Server generierte Dokumente konfiguriert werden soll (Zeichenfolge). Die möglichen Werte sind `'On'`, `'Off'` und `'Email'`. Der Standardwert ist `'Off'`.

```
node[:apache][:serversignature]
```

servertokens

Gibt an, welche Art von Serverversionsinformationen im Antwort-Header enthalten sind (Zeichenfolge):

- `'Full'`: Die vollständigen Informationen. Zum Beispiel Server: Apache/2.4.2 (Unix) PHP/4.2.2 /1.2 MyMod
- `'Prod'`: Produktname. Beispiel, Server: Apache
- `'Major'`: Hauptversion. Beispiel, Server: Apache/2
- `'Minor'`: Haupt- und Nebenversion. Beispiel, Server: Apache/2.4
- `'Min'`: Mindestversion. Beispiel, Server: Apache/2.4.2
- `'OS'`: Version mit Betriebssystem. Beispiel, Server: Apache/2.4.2 (Unix)

Der Standardwert ist `'Prod'`.

```
node[:apache][:servertokens]
```

timeout

Der Zeitraum, den Apache auf E/A wartet (Zahl). Der Standardwert ist `120`.

```
node[:apache][:timeout]
```

traceenable

Gibt an, ob TRACE-Anforderungen aktiviert werden sollen (Zeichenfolge). Die möglichen Werte sind `'On'` und `'Off'`. Der Standardwert ist `'Off'`.

```
node[:apache][:traceenable]
```

user

Der Benutzername (Zeichenfolge). Die Standardwerte lauten wie folgt:

- Amazon Linux und RHEL: `'apache'`
- Ubuntu: `'www-data'`

```
node[:apache][:user]
```

version

Die Apache-Version (Zeichenfolge). Die Standardwerte lauten wie folgt:

- Amazon Linux: `2.2`
- Ubuntu 14.04 LTS: `2.4`
- RHEL: `2.4`

```
node[:apache][:version]
```

worker-Attribute

Diese Attribute geben die Worker-Prozesskonfiguration an.

startservers

Die Anzahl von untergeordneten Serverprozessen, die beim Starten erstellt werden können (Zahl). Der Standardwert ist `4`.

```
node[:apache][:worker][:startservers]
```

maxclients

Die maximale Anzahl von gleichzeitigen Anforderungen, die verarbeitet werden (Zahl). Der Standardwert ist `1024`.

```
node[:apache][:worker][:maxclients]
```

maxsparethreads

Die maximale Anzahl von Leerlaufthreads (Zahl). Der Standardwert ist `192`.

```
node[:apache][:worker][:maxsparethreads]
```

minsparethreads

Die Mindestanzahl von Leerlaufthreads (Zahl). Der Standardwert ist `64`.

```
node[:apache][:worker][:minsparethreads]
```

threadsperchild

Die Anzahl von Threads pro untergeordnetem Prozess (Zahl). Der Standardwert ist `64`.

```
node[:apache][:worker][:threadsperchild]
```

maxrequestsperchild

Die maximale Anzahl von Anforderungen, die ein untergeordneter Serverprozess verarbeiten kann (Zahl). Der Standardwert ist `10000`.

```
node[:apache][:worker][:maxrequestsperchild]
```

Bereitstellungsattribute

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Die integrierte Bereitstellungsattributdatei `deploy.rb` des Rezeptbuchs definiert die folgenden Attribute im `opsworks`-Namespace. Weitere Informationen zu den Bereitstellungsverzeichnissen finden Sie unter Bereitstellungsrezepte. Weitere Informationen zum Überschreiben integrierter Attribute, um benutzerdefinierte Werte anzugeben, finden Sie unter Überschreiben der Attribute.

deploy_keep_releases

Eine globale Einstellung für die Anzahl der App-Bereitstellungen, die AWS OpsWorks Stacks speichert (Anzahl). Der Standardwert ist 5. Dieser Wert bestimmt, wie oft Sie ein Rollback für eine Anwendung ausführen können.

```
node[:opsworks][:deploy_keep_releases]
```

Gruppe

(Nur für Linux) Die `group`-Einstellung für das Bereitstellungsverzeichnis der App (Anwendung). Der Standardwert hängt vom Betriebssystem der Instance ab:

- Für Ubuntu-Instances ist der Standardwert `www-data`.
- Für Amazon Linux- oder RHEL-Instances, die Mitglieder einer Rails-App Server-Schicht sind, die Nginx und Unicorn verwendet, ist der Standardwert. `nginx`
- Für alle anderen Amazon Linux- oder RHEL-Instances lautet der Standardwert `apache`.

```
node[:opsworks][:deploy_user][:group]
```

user

(Nur für Linux) Die `user`-Einstellung für das Bereitstellungsverzeichnis der App (Anwendung). Der Standardwert ist `deploy`.

```
node[:opsworks][:deploy_user][:user]
```

haproxy-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Attribute stehen nur für Linux-Stacks zur Verfügung.

Die `haproxy-Attribute` geben die HAProxy-Serverkonfiguration an. Weitere Informationen finden Sie unter HAProxy-Dokumente. Weitere Informationen zum Überschreiben integrierter Attribute, um benutzerdefinierte Werte anzugeben, finden Sie unter Überschreiben der Attribute.

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

Der Algorithmus, der von einem Load Balancer zum Auswählen eines Servers verwendet wird (Zeichenfolge). Der Standardwert ist `'roundrobin'`. Weitere Optionen:

- 'static-rr'
- 'leastconn'
- 'source'
- 'uri'
- 'url_param'
- 'hdr(name)'
- 'rdp-cookie'
- 'rdp-cookie(name)'

Weitere Informationen zu diesen Argumenten finden Sie unter balance.

```
node[:haproxy][:balance]
```

check_interval

Das Zeitintervall der Zustandsprüfung (Zeichenfolge). Der Standardwert ist `'10s'`.

```
node[:haproxy][:check_interval]
```

client_timeout

Die maximale Zeitspanne, die ein Client inaktiv sein kann (Zeichenfolge). Der Standardwert ist `'60s'`.

```
node[:haproxy][:client_timeout]
```

connect_timeout

Die maximale Zeitspanne, die ein HAProxy wartet, bis eine Verbindung mit einem Server hergestellt wird (Zeichenfolge). Der Standardwert ist `'10s'`.

```
node[:haproxy][:connect_timeout]
```

default_max_connections

Die standardmäßige maximale Anzahl von Verbindungen (Zeichenfolge). Der Standardwert ist `'80000'`.

```
node[:haproxy][:default_max_connections]
```

global_max_connections

Die maximale Anzahl von Verbindungen (Zeichenfolge). Der Standardwert ist `'80000'`.

```
node[:haproxy][:global_max_connections]
```

health_check_method

Die Methode der Zustandsprüfung (Zeichenfolge). Der Standardwert ist `'OPTIONS'`.

```
node[:haproxy][:health_check_method]
```

health_check_url

Der URL-Pfad, der für die Zustandsprüfung des Servers verwendet wird (Zeichenfolge). Der Standardwert ist `'/'`.

```
node[:haproxy][:health_check_url ]
```

queue_timeout

Die maximale Wartezeit für eine kostenlose Verbindung (Zeichenfolge). Der Standardwert ist `'120s'`.

```
node[:haproxy][:queue_timeout]
```

http_request_timeout

Die maximale Zeitspanne, die HAProxy auf eine abgeschlossene HTTP-Anforderung wartet (Zeichenfolge). Der Standardwert ist `'30s'`.

```
node[:haproxy][:http_request_timeout]
```

retries

Die Anzahl von Wiederholungen nach einem Ausfall der Serververbindung (Zeichenfolge). Der Standardwert ist `'3'`.

```
node[:haproxy][:retries]
```

server_timeout

Die maximale Zeitspanne, die ein Client inaktiv sein kann (Zeichenfolge). Der Standardwert ist `'60s'`.

```
node[:haproxy][:server_timeout]
```

stats_url

Der URL-Pfad für die Statistikseite (Zeichenfolge). Der Standardwert ist `'/haproxy?stats'`.

```
node[:haproxy][:stats_url]
```

stats_user

Der Benutzername der Statistikseite (Zeichenfolge). Der Standardwert ist `'opsworks'`.

```
node[:haproxy][:stats_user]
```

Die `maxcon`-Attribute stehen für einen Lastfaktormultiplikator, der zur Berechnung der maximalen Anzahl von Verbindungen herangezogen wird, die HAProxy für Backends zulässt. Nehmen wir zum Beispiel an, Sie haben einen Rails-App-Server auf einer kleinen Instanz mit einem `backend` Wert von 4, was bedeutet, dass AWS OpsWorks Stacks vier Rails-Prozesse für diese Instanz konfiguriert. Wenn Sie den `maxcon_factor_rails_app`-Standardwert 7 verwenden, verarbeitet der HAProxy 28 (4 x 7) Verbindungen mit dem Rails-Server.

maxcon_factor_nodejs_app

Der maxcon-Faktor für einen Node.js-Anwendungsserver (Zahl). Der Standardwert ist `10`.

```
node[:haproxy][:maxcon_factor_nodejs_app]
```

maxcon_factor_nodejs_app_ssl

Der maxcon-Faktor für einen Node.js-Anwendungsserver mit SSL (Zahl). Der Standardwert ist `10`.

```
node[:haproxy][:maxcon_factor_nodejs_app_ssl]
```

maxcon_factor_php_app

Der maxcon-Faktor für eine PHP-Anwendungsserver (Zahl). Der Standardwert ist `10`.

```
node[:haproxy][:maxcon_factor_php_app]
```

maxcon_factor_php_app_ssl

Der maxcon-Faktor für einen PHP-Anwendungsserver mit SSL (Zahl). Der Standardwert ist `10`.

```
node[:haproxy][:maxcon_factor_php_app_ssl]
```

maxcon_factor_rails_app

Der maxcon-Faktor für einen Rails-Anwendungsserver (Zahl). Der Standardwert ist `7`.

```
node[:haproxy][:maxcon_factor_rails_app]
```

maxcon_factor_rails_app_ssl

Der maxcon-Faktor für einen Rails-Anwendungsserver mit SSL (Zahl). Der Standardwert ist `7`.

```
node[:haproxy][:maxcon_factor_rails_app_ssl]
```

maxcon_factor_static

Der maxcon-Faktor für einen statischen Webserver (Zahl). Der Standardwert ist 15.

```
node[:haproxy][:maxcon_factor_static]
```

maxcon_factor_static_ssl

Der maxcon-Faktor für einen statischen Webserver mit SSL (Zahl). Der Standardwert ist 15.

```
node[:haproxy][:maxcon_factor_static_ssl]
```

memcached-Attribute

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Diese Attribute stehen nur für Linux-Stacks zur Verfügung.

Die `memcached-Attribute` geben die Memcached-Serverkonfiguration an. Weitere Informationen zum Überschreiben integrierter Attribute, um benutzerdefinierte Werte anzugeben, finden Sie unter Überschreiben der Attribute.

memory	max_connections	pid_file
port	start_command	stop_command

user		

memory

Der maximal zu verwendende Speicher in MB (Zahl). Der Standardwert ist `512`.

```
node[:memcached][:memory]
```

max_connections

Die maximale Anzahl von Verbindungen (Zeichenfolge). Der Standardwert ist `'4096'`.

```
node[:memcached][:max_connections]
```

pid_file

Die Datei mit der Prozess-ID des Daemons (Zeichenfolge). Der Standardwert ist `'var/run/memcached.pid'`.

```
node[:memcached][:pid_file]
```

port

Der zu überwachende Port (Zahl). Der Standardwert ist `11211`.

```
node[:memcached][:port]
```

start_command

Der Startbefehl (Zeichenfolge). Der Standardwert ist `'/etc/init.d/memcached start'`.

```
node[:memcached][:start_command]
```

stop_command

Der Stoppbefehl (Zeichenfolge). Der Standardwert ist `'/etc/init.d/memcached stop'`.

```
node[:memcached][:stop_command]
```

user

Der Benutzer (Zeichenfolge). Der Standardwert ist `'nobody'`.

```
node[:memcached][:user]
```

mysql-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Attribute stehen nur für Linux-Stacks zur Verfügung.

Die `mysql`-Attribute geben die MySQL-Masterkonfiguration an. Weitere Informationen finden Sie unter Server-Systemvariablen. Weitere Informationen zum Überschreiben integrierter Attribute, um benutzerdefinierte Werte anzugeben, finden Sie unter Überschreiben der Attribute.

basedir	bind_address	clients
conf_dir	confd_dir	datadir
grants_path	mysql_bin	mysqladmin_bin
pid_file	port	root_group
server_root_password	socket	tunable-Attribute

basedir

Das Basisverzeichnis (Zeichenfolge). Der Standardwert ist `'/usr'`.

```
node[:mysql][:basedir]
```

bind_address

Die Adresse, die MySQL überwacht (Zeichenfolge). Der Standardwert ist `'0.0.0.0'`.

```
node[:mysql][:bind_address]
```

clients

Eine Liste der Clients (Zeichenfolgenliste).

```
node[:mysql][:clients]
```

conf_dir

Das Verzeichnis mit der Konfigurationsdatei (Zeichenfolge). Die Standardwerte lauten wie folgt:

- Amazon Linux und RHEL: `'/etc'`
- Ubuntu: `'/etc/mysql'`

```
node[:mysql][:conf_dir]
```

confd_dir

Das Verzeichnis mit zusätzlichen Konfigurationsdateien (Zeichenfolge). Der Standardwert ist `'/etc/mysql/conf.d'`.

```
node[:mysql][:confd_dir]
```

datadir

Das Datenverzeichnis (Zeichenfolge). Der Standardwert ist `'/var/lib/mysql'`.

```
node[:mysql][:datadir]
```

grants_path

Der Speicherort der GRANT-Tabelle (Zeichenfolge). Der Standardwert ist `'/etc/mysql_grants.sql'`.

```
node[:mysql][:grants_path]
```

mysql_bin

Der Speicherort der mysql-Binärdateien (Zeichenfolge). Der Standardwert ist `'/usr/bin/mysql'`.

```
node[:mysql][:mysql_bin]
```

mysqladmin_bin

Der mysqladmin-Speicherort (Zeichenfolge). Der Standardwert ist `'/usr/bin/mysqladmin'`.

```
node[:mysql][:mysqladmin_bin]
```

pid_file

Die Datei mit der Prozess-ID des Daemons (Zeichenfolge). Der Standardwert ist `'/var/run/mysqld/mysqld.pid'`.

```
node[:mysql][:pid_file]
```

port

Der Port, den der Server überwacht (Zahl). Der Standardwert ist `3306`.

```
node[:mysql][:port]
```

root_group

Die Root-Gruppe (Zeichenfolge). Der Standardwert ist `'root'`.

```
node[:mysql][:root_group]
```

server_root_password

Das Root-Passwort des Servers (Zeichenfolge). Der Standardwert ist zufallsgeneriert.

```
node[:mysql][:server_root_password]
```

socket

Der Speicherort der Socket-Datei (Zeichenfolge). Der Standardwert ist `'/var/lib/mysql/mysql.sock'`. Die Standardwerte lauten wie folgt:

- Amazon Linux und RHEL: `'/var/lib/mysql/mysql.sock'`
- Ubuntu: `'/var/run/mysqld/mysqld.sock'`

```
node[:mysql][:socket]
```

tunable-Attribute

Die tunable-Attribute werden zur Performance-Optimierung eingesetzt.

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

Die maximale Anzahl von ausstehenden Anforderungen (Zeichenfolge). Der Standardwert ist `'128'`.

```
node[:mysql][:tunable][:back_log]
```

innodb_additional_mem_pool_size

Die Größe des Pools, den Innodb zum Speichern interner Datenstrukturen verwendet (Zeichenfolge). Der Standardwert ist `'20M'`.

```
node[:mysql][:tunable][:innodb_additional_mem_pool_size]
```

innodb_buffer_pool_size

Die Größe des Innodb-Pufferpools (Zeichenfolge). Der Attributwert wird von AWS OpsWorks Stacks festgelegt und hängt vom Instanztyp ab. Sie können ihn jedoch überschreiben, indem Sie eine benutzerdefinierte JSON-Datei oder eine benutzerdefinierte Attributdatei verwenden.

```
node[:mysql][:tunable][:innodb_buffer_pool_size]
```

innodb_flush_log_at_trx_commit

Gibt an, wie oft Innodb den Protokollpuffer leert (Zeichenfolge). Der Standardwert ist `'2'`. Weitere Informationen finden Sie unter innodb_flush_log_at_trx_commit.

```
node[:mysql][:tunable][:innodb_flush_log_at_trx_commit]
```

innodb_lock_wait_timeout

Die maximale Anzahl von Sekunden, die eine Innodb-Transaktion auf eine Zeilensperre wartet (Zeichenfolge). Der Standardwert ist `'50'`.

```
node[:mysql][:tunable][:innodb_lock_wait_timeout]
```

key_buffer

Die Index-Puffergröße (Zeichenfolge). Der Standardwert ist `'250M'`.

```
node[:mysql][:tunable][:key_buffer]
```

log_slow_queries

Der Speicherort der Protokolldatei für langsame Abfragen (Zeichenfolge). Der Standardwert ist `'/var/log/mysql/mysql-slow.log'`.

```
node[:mysql][:tunable][:log_slow_queries]
```

long_query_time

Die Anzahl von Sekunden, die erforderlich sind, um eine Abfrage als lange Abfrage festzulegen (Zeichenfolge). Der Standardwert ist `'1'`.

```
node[:mysql][:tunable][:long_query_time]
```

max_allowed_packet

Die maximal zulässige Paketgröße (Zeichenfolge). Der Standardwert ist `'32M'`.

```
node[:mysql][:tunable][:max_allowed_packet]
```

max_connections

Die maximale Anzahl von gleichzeitigen Client-Verbindungen (Zeichenfolge). Der Standardwert ist `'2048'`.

```
node[:mysql][:tunable][:max_connections]
```

max_heap_table_size

Die maximale Größe der vom Benutzer erstellten `MEMORY`-Tabellen (Zeichenfolge). Der Standardwert ist `'32M'`.

```
node[:mysql][:tunable][:max_heap_table_size]
```

net_read_timeout

Die Anzahl von Sekunden, die auf mehr Daten aus einer Verbindung gewartet wird (Zeichenfolge). Der Standardwert ist `'30'`.

```
node[:mysql][:tunable][:net_read_timeout]
```

net_write_timeout

Die Anzahl von Sekunden, die gewartet wird, bis ein Block in eine Verbindung geschrieben wird (Zeichenfolge). Der Standardwert ist `'30'`.

```
node[:mysql][:tunable][:net_write_timeout]
```

query_cache_limit

Die maximale Größe einer einzelnen zwischengespeicherten Abfrage (Zeichenfolge). Der Standardwert ist `'2M'`.

```
node[:mysql][:tunable][:query_cache_limit]
```

query_cache_size

Die Cachegröße der Abfrage (Zeichenfolge). Der Standardwert ist `'128M'`.

```
node[:mysql][:tunable][:query_cache_size]
```

query_cache_type

Der Cachetyp der Abfrage (Zeichenfolge). Die möglichen Werte lauten wie folgt:

- `'0'`: Kein Caching und Abrufen von Daten im Cache.
- `'1'`: Cache-Anweisungen, die nicht mit `SELECT SQL_NO_CACHE` beginnen.
- `'2'`: Cache-Anweisungen, die mit `SELECT SQL_CACHE` beginnen.

Der Standardwert ist `'1'`.

```
node[:mysql][:tunable][:query_cache_type]
```

thread_cache_size

Die Anzahl von Client-Threads, die zur Wiederverwendung zwischengespeichert werden (Zeichenfolge). Der Standardwert ist `'8'`.

```
node[:mysql][:tunable][:thread_cache_size]
```

thread_stack

Die Stack-Größe für jeden Thread (Zeichenfolge). Der Standardwert ist `'192K'`.

```
node[:mysql][:tunable][:thread_stack]
```

wait_timeout

Die Anzahl von Sekunden, die auf eine nicht interaktive Verbindung wartet wird. Der Standardwert ist `'180'` (Zeichenfolge).

```
node[:mysql][:tunable][:wait_timeout]
```

table_cache

Die Anzahl von offenen Tabellen (Zeichenfolge). Der Standardwert ist `'2048'`.

```
node[:mysql][:tunable][:table_cache]
```

nginx-Attribute

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Diese Attribute stehen nur für Linux-Stacks zur Verfügung.

Die `nginx`-Attribute geben die Nginx-Konfiguration an. Weitere Informationen finden Sie unter Richtlinienindex. Weitere Informationen zum Überschreiben integrierter Attribute, um benutzerdefinierte Werte anzugeben, finden Sie unter Überschreiben der Attribute.

Binary	dir	gzip
gzip_comp_level	gzip_disable	gzip_http_version

gzip_proxied	gzip_static	gzip_types
gzip_vary	keepalive	keepalive_timeout
log_dir	user	server_names_hash_bucket_size
worker_processes	worker_connections	

Binary

Der Speicherort der Nginx-Binärdateien (Zeichenfolge). Der Standardwert ist `'/usr/sbin/nginx'`.

```
node[:nginx][:binary]
```

dir

Der Speicherort von Dateien, wie z. B. Konfigurationsdateien (Zeichenfolge). Der Standardwert ist `'/etc/nginx'`.

```
node[:nginx][:dir]
```

gzip

Gibt an, ob die gzip-Komprimierung aktiviert ist (Zeichenfolge). Die möglichen Werte sind `'on'` und `'off'`. Der Standardwert ist `'on'`.

> **Warning**
>
> Die Komprimierung kann Sicherheitsrisiken bergen. Um die Komprimierung vollständig zu deaktivieren, legen Sie dieses Attribut wie folgt fest:
>
> ```
> node[:nginx][:gzip] = 'off'
> ```

```
node[:nginx][:gzip]
```

gzip_comp_level

Die Komprimierungsstufe, die zwischen 1 und 9 liegen kann, wobei 1 der geringsten Komprimierung (Zeichenfolge) entspricht. Der Standardwert ist `'2'`.

```
node[:nginx][:gzip_comp_level]
```

gzip_disable

Deaktiviert die gzip-Komprimierung für bestimmte Benutzeragenten (Zeichenfolge). Der Wert ist ein regulärer Ausdruck und der Standardwert lautet `'MSIE [1-6].(?!.*SV1)'`.

```
node[:nginx][:gzip_disable]
```

gzip_http_version

Aktiviert die gzip-Komprimierung für eine bestimmte HTTP-Version (Zeichenfolge). Der Standardwert ist `'1.0'`.

```
node[:nginx][:gzip_http_version]
```

gzip_proxied

Gibt an, ob und wie die Antwort auf Proxy-Anfragen komprimiert werden soll. Die folgenden Werte sind möglich (Zeichenfolge):

- `'off'`: Proxy-Anfragen nicht komprimieren
- `'expired'`: komprimieren, wenn der Expire-Header Caching verhindert
- `'no-cache'`: komprimieren, wenn der Cache-Control-Header auf „no-cache” festgelegt ist
- `'no-store'`: komprimieren, wenn der Cache-Control-Header auf „no-store” festgelegt ist
- `'private'`: komprimieren, wenn der Cache-Control-Header auf „private” festgelegt ist
- `'no_last_modified'`: komprimieren, wenn kein Wert für „Last-Modified” festgelegt ist
- `'no_etag'`: komprimieren, wenn die Anforderung keinen ETag-Header enthält
- `'auth'`: komprimieren, wenn die Anforderung einen Autorisierungs-Header enthält
- `'any'`: alle Proxy-Anfragen komprimieren

Der Standardwert ist `'any'`.

```
node[:nginx][:gzip_proxied]
```

gzip_static

Gibt an, ob das statische gzip-Modul aktiviert ist (Zeichenfolge). Die möglichen Werte sind `'on'` und `'off'`. Der Standardwert ist `'on'`.

```
node[:nginx][:gzip_static]
```

gzip_types

Eine Liste der zu komprimierenden MIME-Typen (Zeichenfolgenliste). Der Standardwert ist `['text/plain', 'text/html', 'text/css', 'application/x-javascript', 'text/xml', 'application/xml', 'application/xml+rss', 'text/javascript']`.

```
node[:nginx][:gzip_types]
```

gzip_vary

Gibt an, ob ein `Vary:Accept-Encoding` Antwort-Header aktiviert werden soll (Zeichenfolge). Die möglichen Werte sind `'on'` und `'off'`. Der Standardwert ist `'on'`.

```
node[:nginx][:gzip_vary]
```

keepalive

Gibt an, ob eine Keepalive-Verbindung aktiviert werden sollen (Zeichenfolge). Die möglichen Werte sind `'on'` und `'off'`. Der Standardwert ist `'on'`.

```
node[:nginx][:keepalive]
```

keepalive_timeout

Die maximale Anzahl von Sekunden, die eine Keepalive-Verbindung offen bleibt (Zahl). Der Standardwert ist 65.

```
node[:nginx][:keepalive_timeout]
```

log_dir

Der Speicherort der Protokolldateien (Zeichenfolge). Der Standardwert ist `'/var/log/nginx'`.

```
node[:nginx][:log_dir]
```

user

Der Benutzer (Zeichenfolge). Die Standardwerte lauten wie folgt:

- Amazon Linux und RHEL: `'www-data'`
- Ubuntu: `'nginx'`

```
node[:nginx][:user]
```

server_names_hash_bucket_size

Die Bucket-Größe für Hash-Tabellen von Servernamen. Diese kann auf `32`, `64` oder `128` festgelegt werden (Zahl). Der Standardwert ist `64`.

```
node[:nginx][:server_names_hash_bucket_size]
```

worker_processes

Die Anzahl von Worker-Prozessen (Zahl). Der Standardwert ist `10`.

```
node[:nginx][:worker_processes]
```

worker_connections

Die maximale Anzahl von Worker-Verbindungen (Zahl). Der Standardwert ist `1024`. Die maximale Anzahl von Clients wird auf `worker_processes * worker_connections` festgelegt.

```
node[:nginx][:worker_connections]
```

opsworks_berkshelf-Attribute

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Attribute stehen nur für Linux-Stacks zur Verfügung.

Die `opsworks_berkshelf`-Attribute geben die Berkshelf-Konfiguration an. Weitere Informationen finden Sie unter Berkshelf. Weitere Informationen zum Überschreiben integrierter Attribute, um benutzerdefinierte Werte anzugeben, finden Sie unter Überschreiben der Attribute.

debug

Gibt an, ob Berkshelf-Debugging-Informationen im Chef-Protokoll enthalten sein sollen (Boolescher Wert). Der Standardwert ist `false`.

```
node['opsworks_berkshelf]['debug']
```

opsworks_java-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Attribute stehen nur für Linux-Stacks zur Verfügung.

Die `opsworks_java`-Attribute geben die Tomcat-Serverkonfiguration an. Weitere Informationen finden Sie unter Referenz zur Apache Tomcat-Konfiguration. Weitere Informationen zum

Überschreiben integrierter Attribute, um benutzerdefinierte Werte anzugeben, finden Sie unter Überschreiben der Attribute.

datasources	java_app_server_version	java_shared_lib_dir
jvm_pkg-Attribute	custom_pkg_location_url_debian	java_home_basedir
custom_pkg_location_url_rhel	use_custom_pkg_location	jvm_options
jvm_version	tomcat-Attribute	

datasources

Eine Gruppe von Attributen, die JNDI-Ressourcennamen definieren (Zeichenfolge). Weitere Informationen zur Verwendung dieses Attributs finden Sie unter Bereitstellen einer JSP-Anwendung mit einer Backend-Datenbank. Der Standardwert ist ein leerer Hash, der mit benutzerdefinierten Zuordnungen zwischen Anwendungskurznamen und JNDI-Namen gefüllt werden kann. Weitere Informationen finden Sie unter Bereitstellen einer JSP-Anwendung mit einer Backend-Datenbank.

```
node['opsworks_java']['datasources']
```

java_app_server_version

Die Version des Java-Anwendungsservers (Zahl). Der Standardwert ist 7. Sie können dieses Attribut überschreiben, um Version 6 anzugeben. Wenn Sie ein nicht standardmäßiges JDK installieren, wird dieses Attribut ignoriert.

```
node['opsworks_java']['java_app_server_version']
```

java_shared_lib_dir

Das Verzeichnis für die für Java freigegebenen Bibliotheken (Zeichenfolge). Der Standardwert ist `/usr/share/java`.

```
node['opsworks_java']['java_shared_lib_dir']
```

jvm_pkg-Attribute

Eine Gruppe von Attributen, die Sie überschreiben können, um ein nicht standardmäßiges JDK zu installieren.

use_custom_pkg_location

Gibt an, ob statt OpenJDK ein benutzerdefiniertes JDK installiert werden soll (Boolescher Wert). Der Standardwert ist `false`.

```
node['opsworks_java']['jvm_pkg']['use_custom_pkg_location']
```

custom_pkg_location_url_debian

Der Speicherort des JDK-Pakets zur Installation auf Ubuntu-Instances (Zeichenfolge). Der Standardwert ist `'http://aws.amazon.com/'`. Dies ist lediglich ein Initialisierungswert ohne Eigenbedeutung. Wenn Sie ein nicht standardmäßiges JDK installieren möchten, müssen Sie dieses Attribut überschreiben und auf die entsprechende URL festlegen.

```
node['opsworks_java']['jvm_pkg']['custom_pkg_location_url_debian']
```

custom_pkg_location_url_rhel

Der Speicherort des JDK-Pakets zur Installation auf Amazon Linux- und RHEL-Instances (Zeichenfolge). Der Standardwert ist `'http://aws.amazon.com/'`. Dies ist lediglich ein Initialisierungswert ohne Eigenbedeutung. Wenn Sie ein nicht standardmäßiges JDK installieren möchten, müssen Sie dieses Attribut überschreiben und auf die entsprechende URL festlegen.

```
node['opsworks_java']['jvm_pkg']['custom_pkg_location_url_rhel']
```

java_home_basedir

Das Verzeichnis, in das das JDK-Paket extrahiert werden soll (Zeichenfolge). Der Standardwert ist `/usr/local`. Sie müssen diese Einstellung für RPM-Pakete nicht angeben. Sie umfassen eine vollständige Verzeichnisstruktur.

```
node['opsworks_java']['jvm_pkg']['java_home_basedir']
```

jvm_options

Die JVM-Befehlszeilenoptionen, mit denen Sie Einstellungen wie die Heap-Größe angeben können (Zeichenfolge). Eine häufig verwendete Gruppe von Optionen ist `-Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC`. Als Standardwert werden keine Optionen verwendet.

```
node['opsworks_java']['jvm_options']
```

jvm_version

Die OpenJDK-Version (Zahl). Der Standardwert ist 7. Sie können dieses Attribut überschreiben, um OpenJDK Version 6 anzugeben. Wenn Sie ein nicht standardmäßiges JDK installieren, wird dieses Attribut ignoriert.

```
node['opsworks_java']['jvm_version']
```

tomcat-Attribute

Eine Gruppe von Attributen, die Sie zur Installation der Tomcat-Standardkonfiguration überschreiben können.

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

Der AJP-Port (Zahl). Der Standardwert ist `8009`.

```
node['opsworks_java']['tomcat]['ajp_port']
```

apache_tomcat_bind_mod

Das Proxy-Modul (Zeichenfolge). Der Standardwert ist `proxy_http`. Sie können dieses Attribut überschreiben, um das AJP-Proxy-Modul `proxy_ajp` anzugeben.

```
node['opsworks_java']['tomcat]['apache_tomcat_bind_mod']
```

apache_tomcat_bind_path

Der Apache-Tomcat-Bindungspfad (Zeichenfolge). Der Standardwert ist `/`. Sie sollten dieses Attribut nicht überschreiben. Wenn Sie den Bindungspfad ändern, funktioniert die Anwendung möglicherweise nicht mehr.

```
node['opsworks_java']['tomcat]['apache_tomcat_bind_path']
```

auto_deploy

Gibt an, ob eine automatische Bereitstellung erfolgen soll (Boolescher Wert). Der Standardwert ist `true`.

```
node['opsworks_java']['tomcat]['auto_deploy']
```

connection_timeout

Die Verbindungszeitüberschreitung in Millisekunden (Zahl). Die Standardwert ist `20000` (20 Sekunden).

```
node['opsworks_java']['tomcat]['connection_timeout']
```

mysql_connector_jar

Die JAR-Datei der MySQL-Konnektorbibliothek (Zeichenfolge). Der Standardwert ist `mysql-connector-java.jar`.

```
node['opsworks_java']['tomcat]['mysql_connector_jar']
```

port

Der Standard-Port (Zahl). Der Standardwert ist `8080`.

```
node['opsworks_java']['tomcat]['port']
```

secure_port

Der sichere Port (Zahl). Der Standardwert ist `8443`.

```
node['opsworks_java']['tomcat]['secure_port']
```

shutdown_port

Der Shutdown-Port (Zahl). Der Standardwert ist `8005`.

```
node['opsworks_java']['tomcat]['shutdown_port']
```

threadpool_max_threads

Die maximale Anzahl von Threads im Thread-Pool (Zahl). Der Standardwert ist `150`.

```
node['opsworks_java']['tomcat]['threadpool_max_threads']
```

threadpool_min_spare_threads

Die Mindestanzahl von Reservethreads im Threadpool (Zahl). Der Standardwert ist `4`.

```
node['opsworks_java']['tomcat]['threadpool_min_spare_threads']
```

unpack_wars

Gibt an, ob WAR-Dateien entpackt werden sollen (Boolescher Wert). Der Standardwert ist `true`.

```
node['opsworks_java']['tomcat]['unpack_wars']
```

uri_encoding

Die URI-Codierung (Zeichenfolge). Der Standardwert ist `UTF-8`.

```
node['opsworks_java']['tomcat]['uri_encoding']
```

use_ssl_connector

Gibt an, ob ein SSL-Konnektor verwendet werden soll (Boolescher Wert). Der Standardwert ist `false`.

```
node['opsworks_java']['tomcat]['use_ssl_connector']
```

use_threadpool

Gibt an, ob ein Threadpool verwendet werden soll (Boolean). Der Standardwert ist `false`.

```
node['opsworks_java']['tomcat]['use_threadpool']
```

userdatabase_pathname

Der Pfadname der Benutzerdatenbank (Zeichenfolge). Der Standardwert ist `conf/tomcat-users.xml`.

```
node['opsworks_java']['tomcat]['userdatabase_pathname']
```

passenger_apache2-Attribute

Important

Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Diese Attribute stehen nur für Linux-Stacks zur Verfügung.

Die `passenger_apache2-Attribute` geben die Phusion Passenger-Konfiguration an. Weitere Informationen finden Sie unter Phusion Passenger-Benutzerhandbuch, Apache-Version. Weitere

Informationen zum Überschreiben integrierter Attribute, um benutzerdefinierte Werte anzugeben, finden Sie unter Überschreiben der Attribute.

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

Gibt an, ob eine Fehlerseite angezeigt werden soll, wenn eine Anwendung nicht gestartet werden kann (Zeichenfolge). Dieses Attribut kann auf „on” oder „off” festgelegt werden. Der Standardwert ist „off”.

```
node[:passenger][:friendly_error_pages]
```

gem_bin

Der Speicherort der Gem-Binärdateien (Zeichenfolge). Der Standardwert ist `'/usr/local/bin/gem'`.

```
node[:passenger][:gem_bin]
```

gems_path

Der Gems-Pfad (Zeichenfolge). Der Standardwert hängt von der Ruby-Version ab. Beispielsweise:

- Ruby Version 1.8: `'/usr/local/lib/ruby/gems/1.8/gems'`
- Ruby Version 1.9: `'/usr/local/lib/ruby/gems/1.9.1/gems'`

```
node[:passenger][:gems_path]
```

high_performance_mode

Gibt an, ob der Hochleistungsmodus von Passenger verwendet werden soll (Zeichenfolge). Die möglichen Werte sind `'on'` und `'off'`. Der Standardwert ist `'off'`.

```
node[:passenger][:high_performance_mode ]
```

root_path

Das Passenger-Stammverzeichnis (Zeichenfolge). Der Standardwert hängt von den Ruby- und Passenger-Versionen ab. In der Chef-Syntax lautet der Wert `"#{node[:passenger][:gems_path]}/passenger-#{passenger[:version]}"`.

```
node[:passenger][:root_path]
```

max_instances_per_app

Die maximale Anzahl von Anwendungsprozessen pro App (Zahl). Der Standardwert ist `0`. Weitere Informationen finden Sie unter. PassengerMaxInstancesPerApp

```
node[:passenger][:max_instances_per_app]
```

max_pool_size

Die maximale Anzahl von Anwendungsprozessoren (Zahl). Der Standardwert ist 8. Weitere Informationen finden Sie unter PassengerMaxPoolSize.

```
node[:passenger][:max_pool_size]
```

max_requests

Die maximale Anzahl von Anforderungen (Zahl). Der Standardwert ist `0`.

```
node[:passenger][:max_requests]
```

module_path

Der Modulpfad (Zeichenfolge). Die Standardwerte lauten wie folgt:

- Amazon Linux und RHEL: `"#{node['apache']['libexecdir']}/mod_passenger.so"`
- Ubuntu: `"#{passenger[:root_path]}/ext/apache2/mod_passenger.so"`

```
node[:passenger][:module_path]
```

pool_idle_time

Die maximale Anzahl von Sekunden, die ein Anwendungsprozess im Leerlauf bleiben kann (Zahl). Der Standardwert lautet `14400` (4 Stunden). Weitere Informationen finden Sie unter PassengerPoolIdleTime.

```
node[:passenger][:pool_idle_time]
```

rails_app_spawner_idle_time

Die maximale Leerlaufzeit für den Rails Application Spawner (Zahl). Wenn dieses Attribut auf null gesetzt ist, erfolgt keine Zeitüberschreitung des Application Spawner. Der Standardwert ist `0`. Weitere Informationen finden Sie unter Erläuterungen der Spawning-Methoden.

```
node[:passenger][:rails_app_spawner_idle_time]
```

rails_framework_spawner_idle_time

Die maximale Leerlaufzeit für den Rails Framework Spawner (Zahl). Wenn dieses Attribut auf null gesetzt ist, erfolgt keine Zeitüberschreitung des Framework Spawner. Der Standardwert ist `0`. Weitere Informationen finden Sie unter Erläuterungen der Spawning-Methoden.

```
node[:passenger][:rails_framework_spawner_idle_time]
```

rails_spawn_method

Die Rails-Spawn-Methode (Zeichenfolge). Der Standardwert ist `'smart-lv2'`. Weitere Informationen finden Sie unter Erläuterungen der Spawning-Methoden.

```
node[:passenger][:rails_spawn_method]
```

ruby_bin

Der Speicherort der Ruby-Binärdateien (Zeichenfolge). Der Standardwert ist `'/usr/local/bin/ruby'`.

```
node[:passenger][:ruby_bin]
```

ruby_wrapper_bin

Der Speicherort des Ruby-Wrapper-Skripts (Zeichenfolge). Der Standardwert ist `'/usr/local/bin/ruby_gc_wrapper.sh'`.

```
node[:passenger][:ruby_wrapper_bin]
```

stat_throttle_rate

Die Rate, mit der Passenger Dateisystemprüfungen durchführt (Zahl). Der Standardwert ist 5. Dies bedeutet, dass die Prüfungen höchstens einmal alle 5 Sekunden ausgeführt werden. Weitere Informationen finden Sie unter PassengerStatThrottleRate .

```
node[:passenger][:stat_throttle_rate]
```

version

Die Version (Zeichenfolge). Der Standardwert ist `'3.0.9'`.

```
node[:passenger][:version]
```

ruby-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Attribute stehen nur für Linux-Stacks zur Verfügung.

Die `ruby`-Attribute geben die Ruby-Version an, die von Anwendungen verwendet wird. Beachten Sie, dass sich die Attributnutzung mit der Einführung von semantischem Versioning in Ruby 2.1 geändert

hat. Weitere Informationen zum Festlegen von Versionen, einschließlich Beispielen, finden Sie unter Ruby-Versionen. Vollständige Informationen darüber, wie AWS OpsWorks Stacks die Ruby-Version bestimmt, finden Sie in der Datei mit den integrierten Attributen ruby.rb. Weitere Informationen zum Überschreiben integrierter Attribute, um benutzerdefinierte Werte anzugeben, finden Sie unter Überschreiben der Attribute.

full_version

Die vollständige Versionsnummer (Zeichenfolge). Sie sollten dieses Attribut nicht überschreiben. Verwenden Sie stattdessen [: opsworks][: ruby_version] und das entsprechende Patch-Version-Attribut, um eine Version anzugeben.

```
[:ruby][:full_version]
```

major_version

Die Hauptversionsnummer (Zeichenfolge). Sie sollten dieses Attribut nicht überschreiben. Verwenden Sie stattdessen [: opsworks][: ruby_version], um die Hauptversion anzugeben.

```
[:ruby][:major_version]
```

minor_version

Die Nebenversionsnummer (Zeichenfolge). Sie sollten dieses Attribut nicht überschreiben. Verwenden Sie stattdessen [: opsworks][: ruby_version], um die Nebenversion anzugeben.

```
[:ruby][:minor_version]
```

patch

Die Patch-Ebene (Zeichenfolge). Dieses Attribut gilt für Ruby Version 2.0.0 und früher. Für spätere Ruby-Versionen verwenden Sie das `patch_version`-Attribut.

```
[:ruby][:patch]
```

Die Patch-Nummer muss mit dem Präfix p angegeben werden. Verwenden Sie z. B. die folgende benutzerdefinierte JSON, um die Patch-Ebene 484 anzugeben.

```
{
  "ruby":{"patch":"p484"}
```

```
}
```

patch_version

Die Patch-Nummer (Zeichenfolge). Dieses Attribut gilt für Ruby Version 2.1 und später. Für frühere Ruby-Versionen, verwenden Sie das `patch`-Attribut.

```
[:ruby][:patch_version]
```

pkgrelease

Die Paketversionsnummer (Zeichenfolge).

```
[:ruby][:pkgrelease]
```

unicorn-Attribute

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Attribute stehen nur für Linux-Stacks zur Verfügung.

Die `unicorn`-Attribute geben die Unicorn-Konfiguration an. Weitere Informationen finden Sie unter Unicorn::Configurator. Weitere Informationen zum Überschreiben integrierter Attribute, um benutzerdefinierte Werte anzugeben, finden Sie unter Überschreiben der Attribute.

accept_filter	backlog	Verzögerung
tcp_nodelay	tcp_nopush	preload_app

timeout	tries	version
worker_processes		

accept_filter

Der Filter „Akzeptieren” `'httpready'` oder `'dataready'` (Zeichenfolge). Der Standardwert ist `'httpready'`.

```
node[:unicorn][:accept_filter]
```

backlog

Die maximale Anzahl von Anforderungen, die die Warteschlange speichern kann (Zahl). Der Standardwert ist `1024`.

```
node[:unicorn][:backlog]
```

Verzögerung

Die Anzahl von Sekunden, die auf einen erneuten Versuch zum Herstellen einer Bindung zum Socket gewartet wird (Zahl). Der Standardwert ist `0.5`.

```
node[:unicorn][:delay]
```

preload_app

Gibt an, ob eine Anwendung im Voraus geladen wird, bevor ein Worker-Prozess verzweigt wird (boolescher Wert). Der Standardwert ist `true`.

```
node[:unicorn][:preload_app]
```

tcp_nodelay

Gibt an, ob der Nagle-Algorithmus für TCP-Sockets deaktiviert werden soll (Boolescher Wert). Der Standardwert ist `true`.

```
node[:unicorn][:tcp_nodelay]
```

tcp_nopush

Gibt an, ob TCP_CORK aktiviert werden soll (Boolescher Wert). Der Standardwert ist `false`.

```
node[:unicorn][:tcp_nopush]
```

timeout

Die maximale Anzahl von Sekunden, die ein Worker für jede Anforderung aufwenden darf (Zahl). Worker, die diesen Zeitüberschreitungswert überschreiten, werden beendet. Der Standardwert ist `60`.

```
node[:unicorn][:timeout]
```

tries

Die maximale Anzahl von Wiederholungen zum Herstellen einer Bindung zum Socket (Zahl). Der Standardwert ist `5`.

```
node[:unicorn][:tries]
```

version

Die Unicorn-Version (Zeichenfolge). Der Standardwert ist `'4.7.0'`.

```
node[:unicorn][:version]
```

worker_processes

Die Anzahl von Worker-Prozessen (Zahl). Der Standardwert ist `max_pool_size`, sofern vorhanden, oder andernfalls `4`.

```
node[:unicorn][:worker_processes]
```

Beheben von Chef 11.10 und früheren Versionen für Linux

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir

empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Weitere Informationen zur Fehlerbehebung finden Sie unter Handbuch zur Fehlersuche und Fehlerbehebung.

Chef-Protokolle für Chef 11.10 und frühere Versionen für Linux

AWS OpsWorks Stacks speichert die Chef-Logs jeder Instanz in ihrem Verzeichnis. `/var/lib/aws/opsworks/chef` Sie benötigen Sudo-Berechtigungen, um auf dieses Verzeichnis zuzugreifen. Das Protokoll für jede Ausführung befindet sich in einer Datei mit dem Namen *YYYY*`-`*MM*`-`*DD*`-`*HH*`-`*MM*`-`*SS*`-`*NN*`.log`.

Weitere Informationen finden Sie hier:

- Anzeige eines Chef-Protokolls mit der Konsole
- Anzeige eines Chef-Protokolls mit CLI oder API
- Interpretieren eines Chef-Protokolls
- Häufige Chef-Protokollfehler

AWS OpsWorks Stacks mit anderen AWS-Services verwenden

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie können Anwendungsserver, die in einem AWS OpsWorks Stacks-Stack ausgeführt werden, eine Vielzahl von AWS-Services verwenden, die nicht direkt in AWS OpsWorks Stacks integriert sind. Beispielsweise können Sie Ihre Anwendungsserver Amazon RDS als Back-End-Datenbank verwenden lassen. Sie können auf diese Services mithilfe des folgenden allgemeinen Musters zugreifen:

1. Erstellen und konfigurieren Sie den AWS-Service mithilfe der AWS-Konsole, der API oder der CLI und zeichnen Sie alle erforderlichen Konfigurationsdaten auf, die die Anwendung für den Zugriff auf den Service benötigt, wie z. B. Hostname oder Port.
2. Erstellen Sie mindestens ein benutzerdefiniertes Rezept zum Konfigurieren der Anwendung, damit sie auf den Service zugreifen kann.

   Das Rezept erhält die Konfigurationsdaten von Attributen der Stack-Konfiguration und JSON-Bereitstellung, die Sie vor der Ausführung der Rezepte mit benutzerdefinierter JSON definieren.
3. Weisen Sie das benutzerdefinierte Rezept dem Deploy-Lebenszyklusereignis auf dem Anwendungsserver-Layer zu.
4. Erstellen Sie ein benutzerdefiniertes JSON-Objekt, das den Konfigurationsdatenattributen entsprechende Werte zuweist, und fügen Sie es der Stack-Konfiguration und der JSON-Bereitstellung hinzu.
5. Stellen Sie die Anwendung für den Stack bereit.

   Die Bereitstellung führt die benutzerdefinierten Rezepte aus, die die Konfigurationsdatenwerte verwenden, die Sie in der benutzerdefinierten JSON zum Konfigurieren der Anwendung definiert haben, damit sie auf den Service zugreifen kann.

In diesem Abschnitt wird beschrieben, wie AWS OpsWorks Stacks-Anwendungsserver auf eine Vielzahl von AWS-Services zugreifen können. Es wird davon ausgegangen, dass Sie bereits mit Chef-Rezeptbüchern vertraut sind und wissen, wie Rezepte Stack- und Konfigurations-JSON-Attribute zum Konfigurieren von Anwendungen verwenden können, in der Regel durch Erstellen von Konfigurationsdateien. Wenn dies nicht der Fall ist, sollten Sie zuerst Cookbooks und Rezepte und Stacks anpassen AWS OpsWorks lesen.

Themen

- Verwenden eines Backend-Datenspeichers
- ElastiCache Redis als In-Memory-Key-Value-Speicher verwenden
- Verwenden eines Amazon S3 S3-Buckets

- AWS CodePipeline Mit AWS OpsWorks Stacks verwenden

Verwenden eines Backend-Datenspeichers

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Anwendungsserver-Stacks enthalten in der Regel einen Datenbankserver zur Bereitstellung eines Back-End-Datenspeichers. AWS OpsWorks Stacks bietet integrierte Unterstützung für MySQL-Server über die MySQL-Ebene und für verschiedene Arten von Datenbankservern über die Amazon Relational Database Service (Amazon RDS) -Schicht. Sie können einen Stack jedoch problemlos so anpassen, dass die Anwendungsserver andere Datenbankserver wie Amazon DynamoDB oder MongoDB verwenden. Dieses Thema beschreibt das grundlegende Vorgehen, um einen Anwendungsserver mit einem AWS-Datenbankserver zu verbinden. Der Stack und die Anwendung aus Erste Schritte mit Chef 11 Linux-Stacks werden verwendet, um zu zeigen, wie ein PHP-Anwendungsserver manuell mit einer RDS-Datenbank verbunden werden kann. Obwohl das Beispiel auf einem Linux-Stack basiert, gelten die grundlegenden Prinzipien auch für Windows-Stacks. Ein Beispiel für die Integration eines MongoDB-Datenbankservers in einen Stack finden Sie unter Deploying MongoDB with. OpsWorks

> **Note**
>
> In diesem Thema wird Amazon RDS als praktisches Beispiel verwendet. Wenn Sie jedoch eine Amazon RDS-Datenbank mit Ihrem Stack verwenden möchten, ist es viel einfacher, eine Amazon RDS-Schicht zu verwenden.

Themen

- So richten Sie eine Datenbankverbindung ein
- So Connect eine Anwendungsserver-Instance mit Amazon RDS

So richten Sie eine Datenbankverbindung ein

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie richten die Verbindung zwischen einem Anwendungsserver und seiner Backend-Datenbank ein, indem Sie ein benutzerdefiniertes Rezept verwenden. Das Rezept konfiguriert den Anwendungsserver wie benötigt, normalerweise es eine Konfigurationsdatei erstellt. Das Rezept bezieht die Verbindungsdaten wie den Host- und Datenbanknamen aus einer Reihe von Attributen in der Stack-Konfiguration und den Bereitstellungsattributen, die AWS OpsWorks Stacks auf jeder Instanz installiert.

Schritt 2 von Erste Schritte mit Chef 11 Linux-Stacks basiert beispielsweise auf einem Stack MyStack mit zwei Ebenen, PHP App Server und MySQL, mit jeweils einer Instanz. Sie stellen eine App namens SimplePHPApp auf der PHP App Server-Instanz bereit, die die Datenbank auf der MySQL-Instanz als Back-End-Datenspeicher verwendet. Wenn Sie die Anwendung bereitstellen, installiert AWS OpsWorks Stacks Stack-Konfigurations- und Bereitstellungsattribute, die die Informationen zur Datenbankverbindung enthalten. Das folgende Beispiel zeigt die Attribute der Datenbankverbindung, dargestellt als JSON:

```
{
  ...
  "deploy": {
    "simplephpapp": {
      ...
      "database": {
        "reconnect": true,
        "password": null,
        "username": "root",
        "host": null,
        "database": "simplephpapp"
        ...
      },
```

```
      ...
    }
  }
}
```

Die Attributwerte werden von AWS OpsWorks Stacks bereitgestellt und entweder generiert oder basieren auf vom Benutzer bereitgestellten Informationen.

Damit SimplePhpApp auf den Datenspeicher zugreifen kann, müssen Sie die Verbindung zwischen dem PHP-Anwendungsserver und der MySQL-Datenbank einrichten, indem Sie dem Deploy-Lifecycle-Ereignis der PHP App Server-Ebene ein benutzerdefiniertes Rezept zuweisen. `appsetup.rb` Wenn Sie SimplePhpApp bereitstellen, wird AWS OpsWorks Stacks ausgeführt`appsetup.rb`, wodurch eine Konfigurationsdatei mit dem Namen erstellt wird, die die Verbindung herstellt`db-connect.php`, wie im folgenden Auszug gezeigt.

```
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
```

Den Variablen, die die Verbindung charakterisieren — `hostuser`, usw. — werden die entsprechenden Werte aus den Deploy-JSON-Attributen zugewiesen. `[:deploy][:app_name][:database]` Der Einfachheit halber wird in diesem Beispiel davon ausgegangen, dass Sie bereits eine Tabelle mit dem Namen `urler` erstellt haben, sodass der Tabellenname durch `[:phpapp][:dbtable]` in der Attributdatei des Rezeptbuchs repräsentiert wird.

Dieses Rezept kann den PHP-Anwendungsserver tatsächlich mit jedem MySQL-Datenbankserver verbinden, nicht nur mit Mitgliedern einer MySQL-Schicht. Um einen anderen MySQL-Server zu verwenden, müssen Sie nur die `[:database]` Attribute auf Werte setzen, die für Ihren Server geeignet sind, was Sie mit benutzerdefiniertem JSON tun können. AWS OpsWorks Stacks integriert diese Attribute und Werte dann in die Stack-Konfiguration und die Bereitstellungsattribute und `appsetup.rb` verwendet sie, um die Vorlage zu erstellen, mit der die Verbindung eingerichtet wird. Weitere Informationen zum Überschreiben der Stack-Konfiguration und der Bereitstellungs-JSON-Datei finden Sie unter Überschreiben der Attribute.

So Connect eine Anwendungsserver-Instance mit Amazon RDS

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Abschnitt wird beschrieben, wie Sie das Formular so anpassen können MyStack Erste Schritte mit Chef 11 Linux-Stacks, dass der PHP-Anwendungsserver eine Verbindung zu einer RDS-Instance herstellt.

Themen

- Eine Amazon RDS MySQL-Datenbank erstellen
- Anpassen des Stacks für die Verbindung mit der RDS-Datenbank

Eine Amazon RDS MySQL-Datenbank erstellen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Jetzt sind Sie bereit, mit dem Launch DB Instance Wizard der Amazon RDS-Konsole eine RDS-Datenbank für das Beispiel zu erstellen. Das folgende Verfahren ist eine kurze Zusammenfassung der wichtigsten Details. Eine detaillierte Beschreibung, wie Sie eine Datenbank erstellen, finden Sie unter Erste Schritte mit Amazon RDS.

Um die Amazon RDS-Datenbank zu erstellen

1. Wenn Sie zum ersten Mal eine RDS-Datenbank erstellen, klicken Sie auf Get Started Now (Legen Sie gleich los). Klicken Sie im Navigationsbereich auf RDS Dashboard (RDS-Dashboard) und dann auf Launch a DB Instance (DB-Instance starten).
2. Wählen Sie die MySQL Community Edition als DB-Instance aus.
3. Wählen Sie für Do you plan to use this database for production purposes? (Möchten Sie diese Datenbank für Produktionszwecke verwenden?) die Option No, this instance... (Nein, diese Instance...) aus, das ist für dieses Beispiel ausreichend. Für Produktionszwecke wäre die Option Yes, use Multi-AZ Deployment... (Ja, Multi-AZ-Bereitstellung verwenden...) wahrscheinlich besser geeignet. Klicken Sie auf Next Step (Nächster Schritt).
4. Legen Sie auf der Seite Specify DB Details (DB-Details angeben) die folgenden Einstellungen fest:
   - DB Instance Class (DB-Instance-Klasse): db.t2.micro (db.t2.micro)
   - Multi-AZ Deployment (Multi-AZ-Bereitstellung): No (Nein)
   - Allocated Storage (Zugewiesener Speicher): **5** GB
   - DB instance identifier (DB-Instance-Kennung): **`rdsexample`**
   - Master Username (Hauptbenutzername): **`opsworksuser`**.

- Master Password (Hauptpasswort): Geben Sie ein geeignetes Passwort ein und notieren Sie es für eine spätere Nutzung.

Akzeptieren Sie die Standardeinstellungen für die anderen Optionen und klicken Sie dann auf Next Step (Nächster Schritt).

5. Legen Sie auf der Seite Configure Advanced Settings (Erweiterte Einstellungen konfigurieren) die folgenden Einstellungen fest:

   - Wählen Sie im Abschnitt Network & Security (Netzwerk und Sicherheit) für VPC Security Group(s) (VPC-Sicherheitsgruppe(n)) die Option phpsecgroup (VPC) aus.
   - Geben Sie im Abschnitt Database Options (Datenbankoptionen) für Database Name (Datenbankname) **`rdsexampledb`** ein.
   - Legen Sie im Abschnitt Backup die Backup Retention Period (Aufbewahrungszeitraum für Backups) für die Zwecke dieser Anleitung auf 0 fest.

   Akzeptieren Sie die Standardeinstellungen für die anderen Optionen und klicken Sie dann auf Launch DB Instance (DB-Instance starten).

6. Wählen Sie View Your DB Instances (DB-Instances anzeigen) aus, um eine Liste der DB-Instances anzuzeigen.
7. Wählen Sie die rdsexample-Instance in der Liste aus und klicken Sie auf den Pfeil, sodass der Instance-Endpunkt und andere Details angezeigt werden. Notieren Sie den Endpunkt für die spätere Verwendung. Dies sieht etwa so aus: `rdsexample.c6c8mntzhgv0.us-west-2.rds.amazonaws.com:3306`. Notieren Sie sich nur den DNS-Namen. Die Portnummer werden Sie nicht benötigen.
8. Verwenden Sie ein Tool wie MySQL Workbench zum Erstellen einer Tabelle mit dem Namen `urler` in der `rdsexampledb`-Datenbank, indem Sie folgenden SQL-Befehl benutzen:

```
CREATE TABLE urler(id INT UNSIGNED NOT NULL AUTO_INCREMENT,author VARCHAR(63) NOT
 NULL,message TEXT,PRIMARY KEY (id))
```

Anpassen des Stacks für die Verbindung mit der RDS-Datenbank

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sobald Sie eine RDS-Instanz erstellt haben, die als Back-End-Datenbank für den PHP-Anwendungsserver verwendet werden soll, können Sie diese anpassen. MyStack Erste Schritte mit Chef 11 Linux-Stacks

Verbinden des PHP-Anwendungsservers mit einer RDS-Datenbank

1. Öffnen Sie die AWS OpsWorks Stacks-Konsole und erstellen Sie einen Stack mit einer PHP-App-Serverebene, die eine Instanz enthält, und stellen Sie SimplePHPApp bereit, wie unter beschrieben. Erste Schritte mit Chef 11 Linux-Stacks Dieser Stack verwendet Version1 von SimplePHPApp, die keinen Datenbankverbindung benutzt.
2. Aktualisieren Sie die Stack-Konfiguration für die Nutzung der benutzerdefinierten Rezeptbücher, die das `appsetup.rb`-Rezept und verwandte Vorlage- und Attributdateien enthalten.

    1. Legen Sie Use custom Chef Cookbooks (Benutzerdefinierte Chef-Rezeptbücher verwenden) auf Yes (Ja) fest,
    2. Legen Sie Repository type (Repository-Typ) auf Git und Repository URL (Repository-URL) auf `git://github.com/amazonwebservices/opsworks-example-cookbooks.git` fest.
3. Fügen Sie Folgendes dem Feld Custom Chef JSON (Benutzerdefinierte JSON-Chef-Dateien) des Stacks hinzu, um die RDS-Verbindungsdaten den `[:database]`-Attributen zuzuweisen, die `appsetup.rb` verwendet, um die Konfigurationsdatei zu erstellen.

```
{
  "deploy": {
    "simplephpapp": {
      "database": {
        "username": "opsworksuser",
        "password": "your_password",
```

```
        "database": "rdsexampledb",
        "host": "rds_endpoint",
        "adapter": "mysql"
      }
    }
  }
}
```

Verwenden Sie die folgenden Attributwerte:

- username: Der Hauptbenutzername, den Sie beim Erstellen der RDS-Instance festgelegt haben.

  Dieses Beispiel verwendet `opsworksuser`.
- password: Das Hauptpasswort, das Sie beim Erstellen der RDS-Instance festgelegt haben.

  Geben Sie das Passwort ein, das Sie angegeben haben.
- database: Die Datenbank, die Sie beim Erstellen der RDS-Instance angelegt haben.

  Dieses Beispiel verwendet `rdsexampledb`.
- host: Der RDS-Instance-Endpunkt, den Sie von der RDS-Konsole bekommen haben, als Sie die Instance im vorherigen Abschnitt erstellt haben. Schließen Sie nicht die Portnummer ein.
- adapter: Der Adapter.

  Die RDS-Instance für dieses Beispiel verwendet MySQL, sodass adapter auf `mysql` festgelegt ist. Im Gegensatz zu anderen Attributen wird adapter nicht von `appsetup.rb` verwendet. Es wird stattdessen vom integrierten Konfigurationsrezept der PHP App Server-Ebene verwendet, um eine andere Konfigurationsdatei zu erstellen.

4. Bearbeiten Sie die SimplePHPApp-Konfiguration, um eine Version von SimplePHPApp anzugeben, die eine Backend-Datenbank verwendet. Gehen Sie dazu folgendermaßen vor:

   - Document root: Legen Sie diese Option auf `web` fest.
   - Branch/Revision: Legen Sie diese Option auf `version2` fest.

   Lassen Sie die übrigen Optionen unverändert.
5. Bearbeiten Sie die PHP App Server-Ebene, um die Datenbankverbindung einzurichten, indem Sie sie `phpapp::appsetup` zu den Deploy-Rezepten der Ebene hinzufügen.
6. Stellen Sie die neue SimplePHPApp-Version bereit.

7. Wenn SimplePHPApp bereitgestellt ist, führen Sie die Anwendung aus, indem Sie die Seite Instances aufrufen und auf die öffentliche IP-Adresse der Instance "php-app1" klicken. Sie sollten die folgende Seite in Ihrem Browser sehen, auf der Sie Text eingeben und in der Datenbank speichern können.

> **Note**
>
> Wenn Ihr Stack eine MySQL-Schicht hat, weist AWS OpsWorks Stacks den Attributen automatisch die entsprechenden Verbindungsdaten zu. `[:database]` Wenn Sie dem Stack jedoch ein benutzerdefiniertes JSON-Objekt zuweisen, das andere `[:database]`-Werte festlegt, überschreiben sie diese Standardwerte. Da die `[:deploy]` Attribute auf jeder Instanz installiert sind, verwenden alle Rezepte, die von den `[:database]` Attributen abhängen, die benutzerdefinierten Verbindungsdaten, nicht die Daten der MySQL-Schicht für. Wenn Sie möchten, dass ein bestimmter Anwendungsserver-Layer die benutzerdefinierten Verbindungsdaten verwendet, weisen Sie dem Bereitstellungsereignis des Layers das benutzerdefinierte JSON-Objekt zu und schränken Sie die Bereitstellung zu diesem Layer ein. Weitere Informationen zur Verwendung von Bereitstellungsattributen finden Sie unter Bereitstellen von Anwendungen. Weitere Informationen zum Überschreiben von integrierten Attributen des AWS OpsWorks Stacks finden Sie unter Überschreiben der Attribute.

ElastiCache Redis als In-Memory-Key-Value-Speicher verwenden

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Dieses Thema basiert auf einem Linux-Stack, aber Windows-Stacks können auch Amazon ElastiCache (ElastiCache) verwenden. Ein Beispiel für die Verwendung ElastiCache mit einer Windows-Instance finden ElastiCache Sie unter ASP.NET-Sitzungsspeicher.

Sie können die Leistung von Anwendungsservern häufig verbessern, indem Sie einen Cacheserver verwenden, der einen speicherinternen Schlüsselwertspeicher für kleine Datenelemente wie Zeichenfolgen bereitstellt. Amazon ElastiCache ist ein AWS-Service, mit dem Sie ganz einfach Caching-Unterstützung für Ihren Anwendungsserver bereitstellen können, indem Sie entweder die Memcached - oder Redis-Caching-Engines verwenden. AWS OpsWorks Stacks bietet integrierte Unterstützung für Memcached. Wenn Redis jedoch besser zu Ihren Anforderungen passt, können Sie Ihren Stack so anpassen, dass Ihre Anwendungsserver Redis verwenden. ElastiCache

In diesem Thema werden Sie anhand eines Rails-Anwendungsservers durch den grundlegenden Prozess der Bereitstellung von ElastiCache Redis-Caching-Unterstützung für Linux-Stacks geführt. Dabei wird davon ausgegangen, dass Sie bereits über eine geeignete Ruby on Rails-Anwendung verfügen. Weitere Informationen finden Sie ElastiCache unter Was ist Amazon ElastiCache? .

Themen

- Schritt 5: Hinzufügen von JSON-Zugriffsinformationen zur Stack-Konfiguration
- Schritt 6: Bereitstellen und Ausführen der Anwendung

Schritt 1: Erstellen Sie einen ElastiCache Redis-Cluster

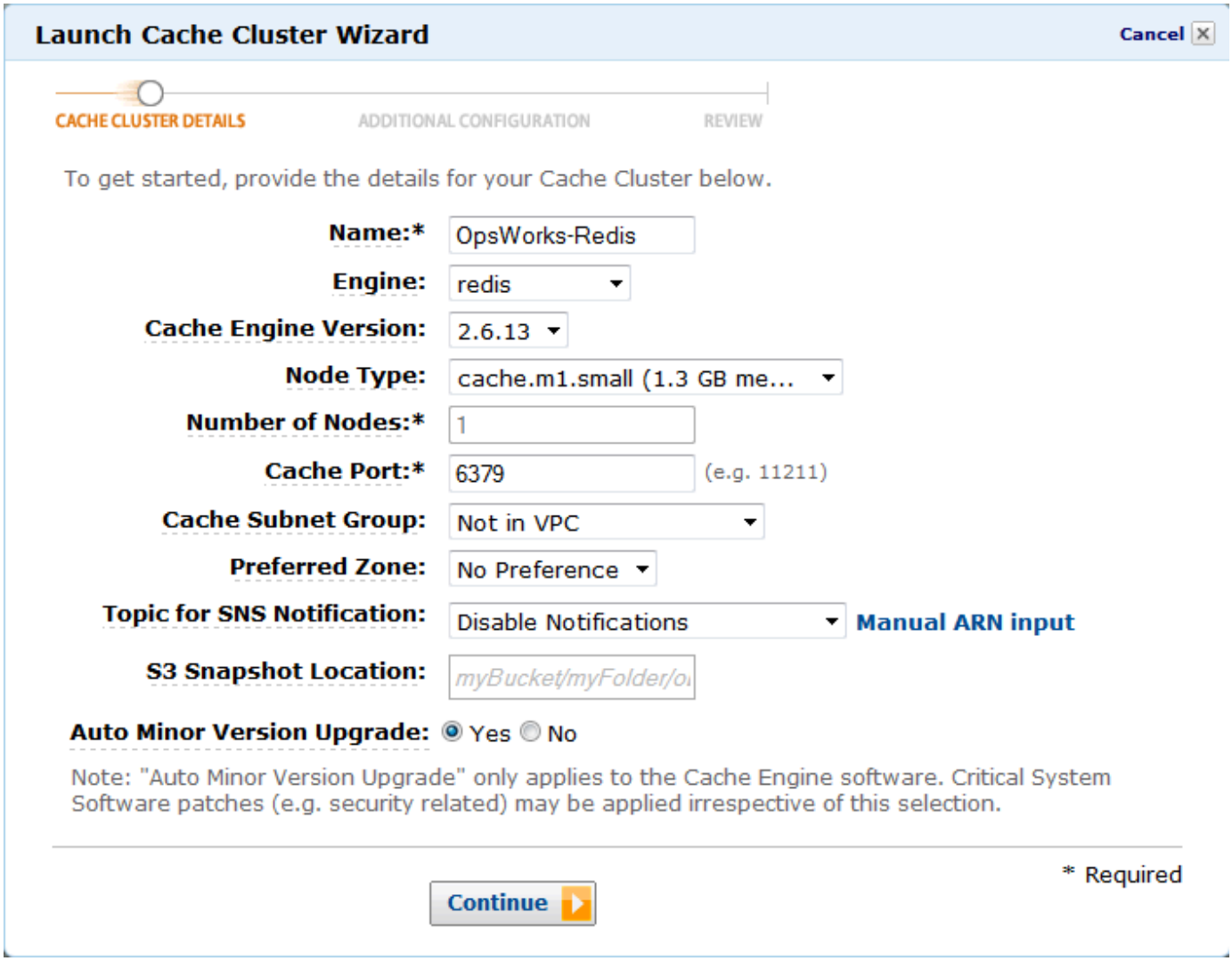
> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie müssen zuerst einen Amazon ElastiCache Redis-Cluster mithilfe der ElastiCache Konsole, API oder CLI erstellen. Im Folgenden wird beschrieben, wie Sie die Konsole verwenden, um ein Cluster zu erstellen.

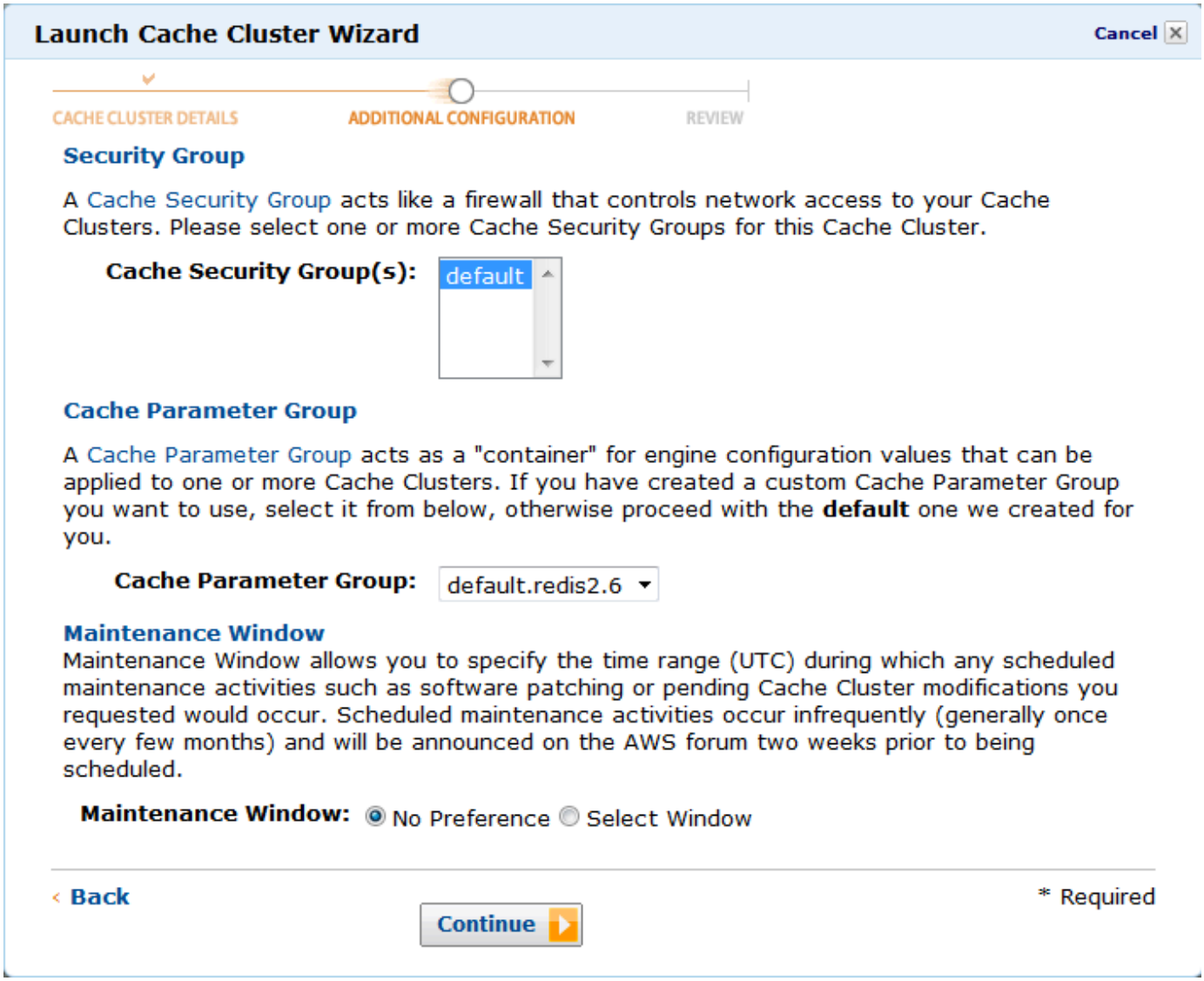
Um einen ElastiCache Redis-Cluster zu erstellen

1. Gehen Sie zur ElastiCacheKonsole und klicken Sie auf Cache-Cluster starten, um den Cache-Cluster-Assistenten zu starten.
2. Führen Sie auf der Seite zu den Cache-Cluster-Details die folgenden Schritte aus:
   - Geben Sie Name als Namen für Ihren Cache-Server ein.

     In diesem Beispiel wird OpsWorks -Redis verwendet.
   - Legen Sie Engine auf redis fest.
   - Legen Sie Topic for SNS Notification (Thema der SNS-Benachrichtigung) auf Disable Notifications (Benachrichtigungen deaktivieren) fest.
   - Übernehmen Sie die Standardwerte für die restlichen Einstellungen und klicken Sie auf Continue (Weiter).



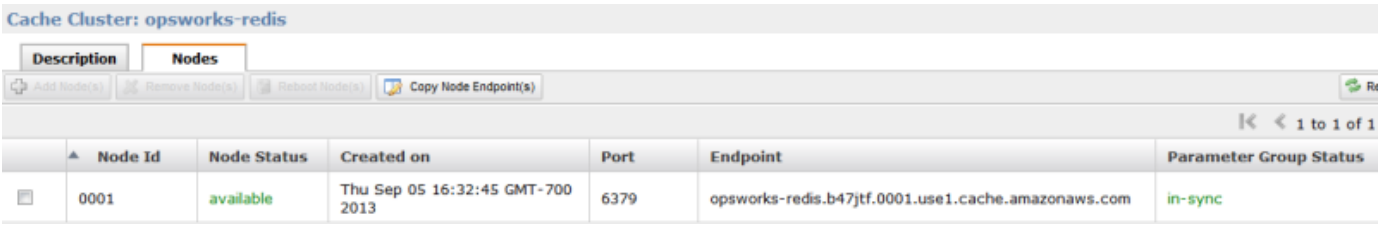
3. Übernehmen Sie auf der Seite Additional Configuration (Zusätzliche Konfiguration) alle Standardinformationen und klicken Sie auf Continue (Weiter).



4. Klicken Sie auf Launch Cache Cluster (Cache-Cluster starten), um den Cluster zu erstellen.

> **Important**
>
> Die Standard-Cache-Sicherheitsgruppe ist für dieses Beispiel ausreichend, für die Produktion sollten Sie jedoch eine für Ihre Umgebung geeignete Sicherheitsgruppe erstellen. Weitere Informationen finden Sie unter Verwalten von Cache-Sicherheitsgruppen.

5. Nachdem der Cluster gestartet wurde, klicken Sie zuerst auf den Namen, um die Detailseite anzuzeigen, und dann auf die Registerkarte Nodes (Knoten). Notieren Sie die Cluster-Werte Port und Endpoint (Endpunkt) zur späteren Verwendung.



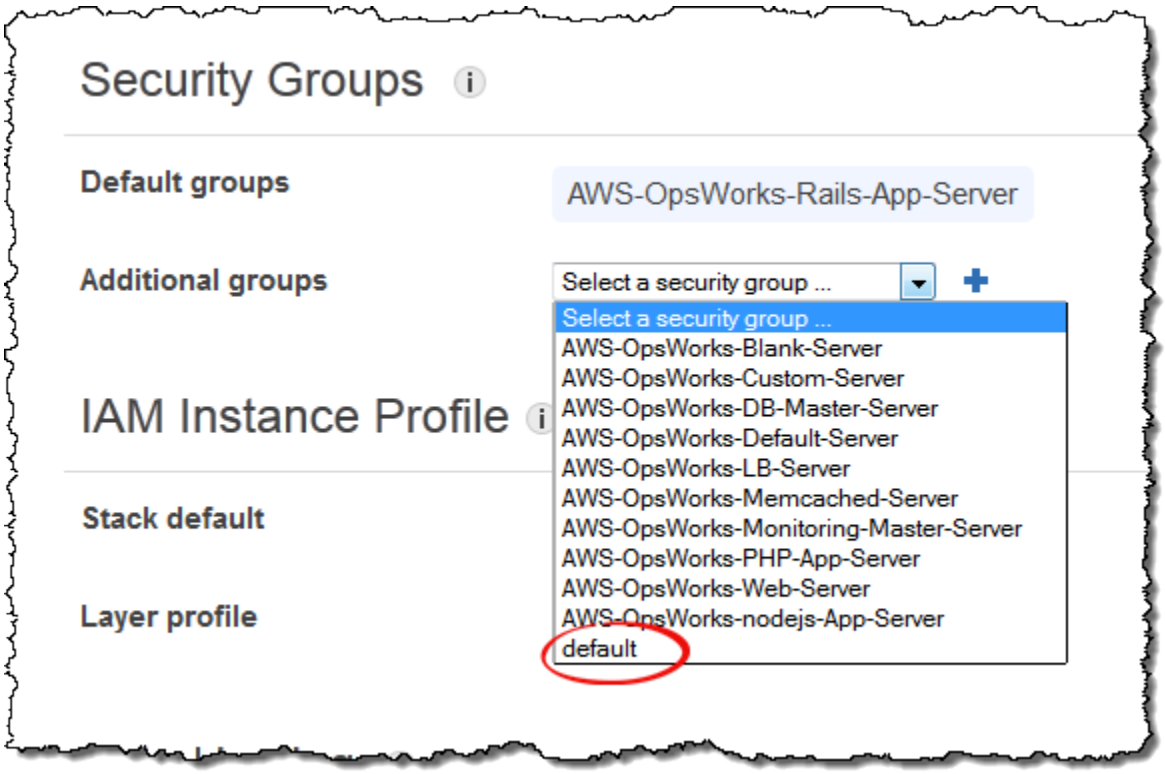
Schritt 2: Einrichten eines Rails-Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie müssen nicht nur einen Stack erstellen, der eine Rails App Server-Schicht unterstützt, sondern auch die Sicherheitsgruppen der Ebene konfigurieren, damit der Rails-Server ordnungsgemäß mit dem Redis-Server kommunizieren kann.

So richten Sie einen Stack ein

1. Erstellen Sie einen neuen Stack — benannt nach diesem **`RedisStack`** Beispiel — und fügen Sie einen Rails App Server-Layer hinzu. Sie können für beide die Standardeinstellungen benutzen. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks und Eine OpsWorks Ebene erstellen .
2. Klicken Sie auf der Seite Layers für Rails App Server auf Sicherheit und dann auf Bearbeiten.
3. Gehen Sie zum Abschnitt Sicherheitsgruppen und fügen Sie die Sicherheitsgruppe des ElastiCache Clusters zu Zusätzliche Gruppen hinzu. Wählen Sie für dieses Beispiel die Sicherheitsgruppe default (Standard) aus, klicken Sie auf +, um sie zum Layer hinzuzufügen, und dann auf Save (Speichern), um die neue Konfiguration zu speichern.



4. Fügen Sie dem Rails App Server-Layer eine Instanz hinzu und starten Sie sie. Weitere Informationen zum Hinzufügen und Starten von Instances finden Sie unter Hinzufügen einer Instance zu einem Layer.

Schritt 3: Erstellen und Bereitstellen eines benutzerdefinierten Rezeptbuchs

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Der Stack ist so noch nicht funktionsfähig. Sie müssen Ihre Anwendung für den Zugriff auf den Redis-Server aktivieren. Die flexibelste Lösung ist die Ablage einer YAML-Datei mit den Zugriffsinformationen im Unterorder `config` der Anwendung. Die Anwendung kann anschließend die Informationen aus der Datei abrufen. Mit diesem Ansatz können Sie die Verbindungsinformationen ändern, ohne die Anwendung neu schreiben oder erneut bereitstellen zu müssen. In diesem Beispiel

sollte die Datei wie folgt benannt werden `redis.yml` und den Hostnamen und Port des ElastiCache Clusters enthalten:

```
host: cache-cluster-hostname
port: cache-cluster-port
```

Sie könnten diese Datei manuell auf Ihre Server kopieren, aber ein besserer Ansatz besteht darin, ein Chef-Rezept zum Generieren der Datei zu implementieren und AWS OpsWorks Stacks das Rezept auf jedem Server ausführen zu lassen. Chef-Rezepte sind spezielle Ruby-Anwendungen, mit denen AWS OpsWorks Stacks Aufgaben auf Instanzen wie das Installieren von Paketen oder das Erstellen von Konfigurationsdateien ausführt. Die Rezepte sind in einem Rezeptbuch gebündelt, das mehrere Rezepte und zugehörige Dateien enthalten kann, wie z. B. Vorlagen für Konfigurationsdateien. Das Kochbuch befindet sich in einem Repository, z. B. GitHub, und muss eine Standardverzeichnisstruktur haben. Wenn Sie noch nicht über ein benutzerdefiniertes Rezeptbuch-Repository verfügen, finden Sie unter Rezeptbuch-Repositorys Informationen, wie Sie es erstellen können.

Fügen Sie für dieses Beispiel ein Rezeptbuch mit dem Namen `redis-config` zu Ihrem Rezeptbuch-Repository mit folgendem Inhalt hinzu:

```
my_cookbook_repository
  redis-config
    recipes
      generate.rb
    templates
      default
        redis.yml.erb
```

Das Verzeichnis `recipes` enthält ein Rezept mit dem Namen `generate.rb`, mit der die Konfigurationsdatei der Anwendung aus `redis.yml.erb` folgendermaßen erzeugt wird:

```
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
```

Das Rezept hängt von Daten aus dem JSON-Objekt AWS OpsWorks Stacks-Stack-Konfiguration und Bereitstellung ab, das auf jeder Instanz installiert ist und detaillierte Informationen über den Stack und alle bereitgestellten Apps enthält. Der `deploy`-Knoten des Objekts hat folgende Struktur:

```
{
   ...
  "deploy": {
    "app1": {
      "application" : "short_name",
      ...
    }
    "app2": {
      ...
    }
    ...
  }
}
```

Der Bereitstellungsknoten enthält für jede bereitgestellte Anwendungen jeweils einen Satz eingebetteter JSON-Objekte, der mit der Kurzbezeichnung der Anwendung benannt ist. Jedes

Objekt enthält eine Gruppe von Attributen, die die Konfiguration der Anwendung definieren, wie beispielsweise das Dokument-Stammverzeichnis und den Anwendungstyp. Eine Liste der Bereitstellungsattribute finden Sie unter Bereitstellungsattribute. Die Werte der Stack-Konfiguration und JSON-Bereitstellung können in den Rezepten mithilfe der Chef-Attributsyntax dargestellt werden. Beispielsweise stellt `[:deploy][:app1][:application]` die Kurzbezeichnung der Anwendung App1 dar.

Für jede Anwendung in `[:deploy]` führt das Rezept den zugehörigen Programmblock aus, wobei `deploy_config` das Anwendungsattribut darstellt. Das Rezept setzt `app_root` auf das Stammverzeichnis der Anwendung, `[:deploy][:app_name][:deploy_to]/current`. Anschließend wird mithilfe einer Chef-Vorlagenressource eine Konfigurationsdatei `redis.yml.erb` erstellt und im Verzeichnis `app_root/config` gespeichert.

Konfigurationsdateien werden in der Regel mit Vorlagen erstellt, in denen die meisten Einstellungen von Chef-Attributen definiert sind. Mit Attributen können Sie unter Verwendung eines benutzerdefinierten JSON-Objekts Einstellungen ändern, wie nachfolgend beschrieben, anstatt die Vorlagedatei neu zu schreiben. Die Vorlage `redis.yml.erb` enthält Folgendes:

```
host: <%= @redis[:host] %>
port: <%= @redis[:port] || 6379 %>
```

Die Elemente **<%... %>** sind Platzhalter für einen Attributwert.

- `<%= @redis[:host] %>` stellt den Wert `redis[:host]` dar, der dem Hostnamen des Cache-Clusters entspricht.
- `<%= @redis[:port] || 6379 %>` gibt den Wert von `redis[:port]` an, oder wenn dieses Attribut nicht definiert ist, lautet der Standard-Port 6379.

Die Ressource `template` funktioniert folgendermaßen:

- `source` und `cookbook` geben jeweils Namen für die Vorlage und das Rezeptbuch an.
- `mode`, `group`und `owner` geben der Konfigurationsdatei dieselben Zugriffsrechte wie die Anwendung.
- Im Abschnitt `variables` wird die in der Vorlage verwendete Variable `@redis` auf den Attributwert `[:redis]` der Anwendung festgelegt.

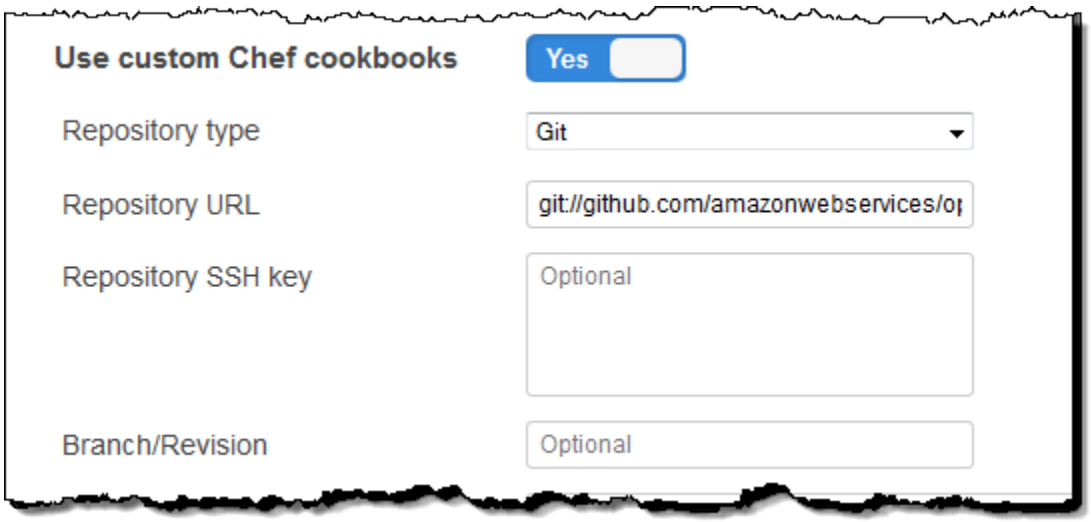
Die `[:redis]`-Attributwerte werden unter Verwendung eines benutzerdefinierten JSON-Objekts festgelegt, wie nachfolgend beschrieben. Es handelt sich nicht um Standard-Anwendungsattribute.

- Der Befehl `not_if` stellt sicher, dass das Rezept keine Konfigurationsdatei erstellt, wenn bereits eine vorhanden ist.

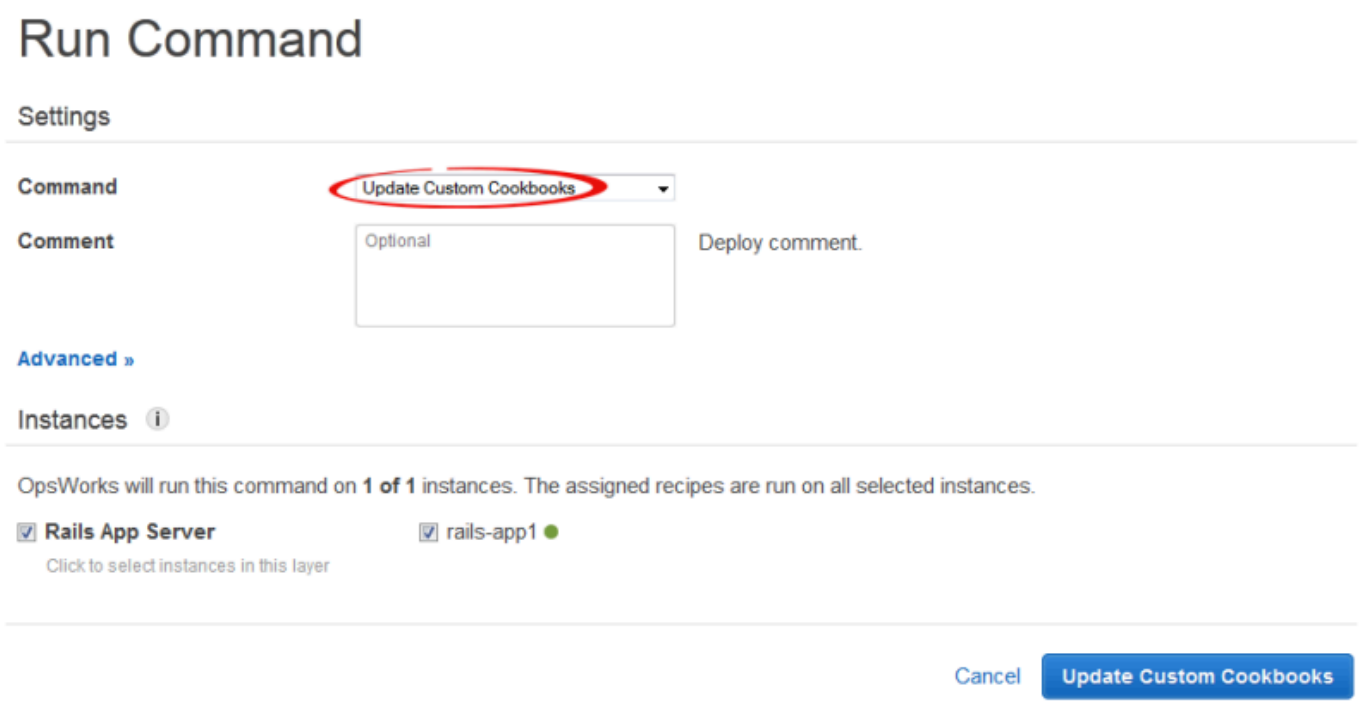
Nachdem Sie das Rezeptbuch erstellt haben, müssen Sie es für den Rezeptbuch-Cache jeder Instance bereitstellen. Mit diese Operation wird nicht das Rezept ausgeführt, sondern lediglich das Rezeptbuch in den Stack-Instances installiert. Normalerweise führen Sie ein Rezept aus, indem Sie es einem Lebenszyklusereignis eines Layers zuweisen, wie nachfolgend beschrieben.

Bereitstellen Ihres benutzerdefinierten Rezeptbuchs

1. Klicken Sie auf der AWS OpsWorks Stacks-Stack-Seite auf Stack-Einstellungen und dann auf Bearbeiten.
2. Legen Sie im Abschnitt Configuration Management (Konfigurationsverwaltung) die Option Use custom Chef cookbooks (Verwenden von benutzerdefinierten Chef-Rezeptbüchern) auf Yes (Ja) fest, geben Sie die Repository-Information des Rezeptbuchs an und klicken Sie auf Save (Speichern), um die Stack-Konfiguration zu aktualisieren.



3. Klicken Sie auf der Seite Stack auf Run Command, wählen Sie den Stack-Befehl Update Custom Cookbooks (Aktualisieren benutzerdefinierter Rezeptbücher) aus und klicken Sie auf Update Custom Cookbooks (Aktualisieren benutzerdefinierter Rezeptbücher), um das neue Rezeptbuch im Rezeptbuch-Cache der Instance zu installieren.



Wenn Sie Ihr Rezeptbuch ändern, führen Sie einfach Update Custom Cookbooks (Aktualisieren benutzerdefinierter Rezeptbücher) erneut aus, um die aktualisierte Version zu installieren. Weitere Informationen zu diesem Verfahren finden Sie unter Installieren von benutzerdefinierten Rezeptbüchern.

Schritt 4: Ordnen Sie das Rezept einem LifeCycle Ereignis zu

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

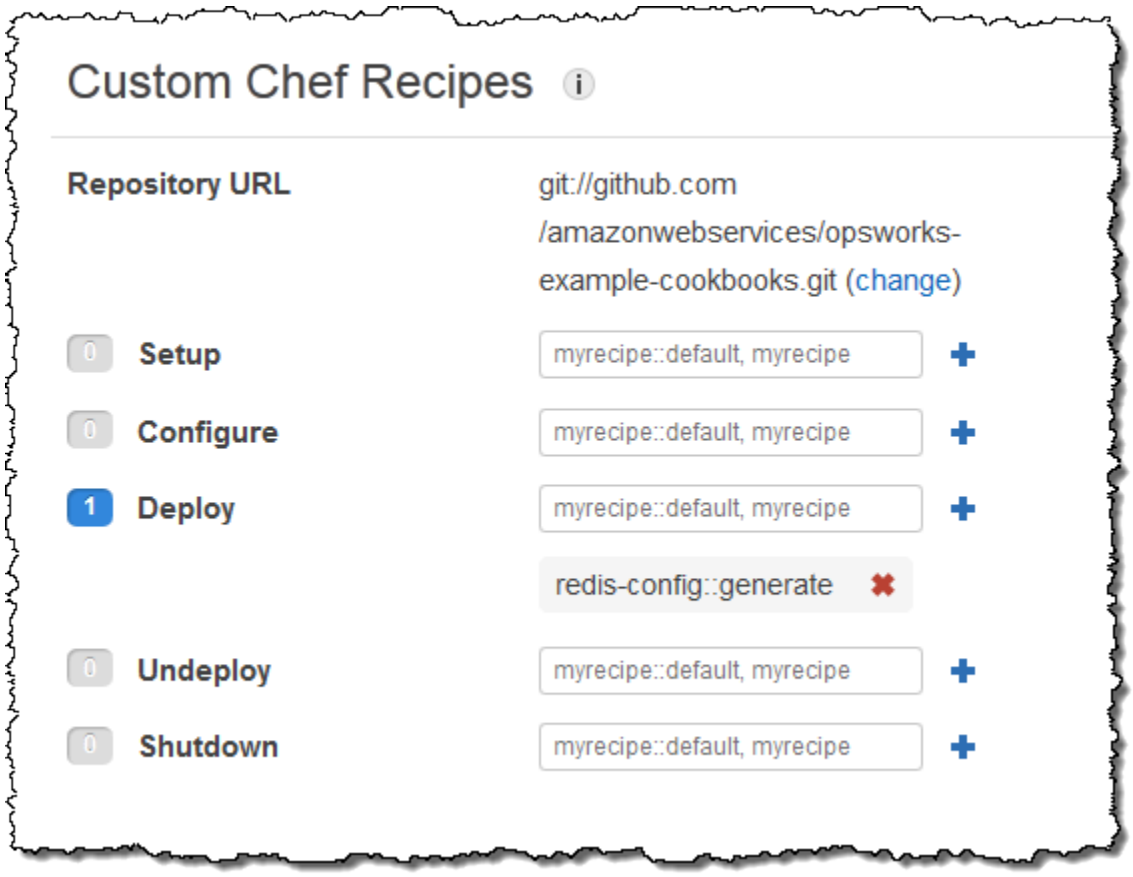
Du kannst benutzerdefinierte Rezepte manuell ausführen, aber der beste Ansatz ist normalerweise, sie von AWS OpsWorks Stacks automatisch ausführen zu lassen. Jeder Ebene ist ein Satz integrierter Rezepte zugeordnet, denen jeweils fünf Lebenszyklusereignisse zugewiesen sind: Setup, Configure, Deploy, Deployment und Shutdown. Jedes Mal, wenn für eine Instance ein Ereignis stattfindet, führt AWS OpsWorks Stacks die zugewiesenen Rezepte für alle Instance-Layer aus,

die die entsprechenden Aufgaben verwalten. Wenn eine Instanz beispielsweise den Startvorgang abgeschlossen hat, löst AWS OpsWorks Stacks ein Setup-Ereignis aus. In diesem Fall werden die zugehörigen Einrichtungsrezepte des Layers ausgeführt, die in der Regel für Aufgaben wie die Installation und Konfiguration von Paketen zuständig sind.

Sie können AWS OpsWorks Stacks ein benutzerdefiniertes Rezept für die Instanzen einer Ebene ausführen lassen, indem Sie das Rezept dem entsprechenden Lebenszyklusereignis zuweisen. In diesem Beispiel sollten Sie das `generate.rb` Rezept dem Deploy-Ereignis der Rails-App Serverebene zuweisen. AWS OpsWorks Stacks führt es dann beim Start, nach Abschluss der Setup-Rezepte und jedes Mal, wenn Sie eine App bereitstellen, auf den Instanzen des Layers aus. Weitere Informationen finden Sie unter Automatisches Ausführen von Rezepten.

Um dem Deploy-Ereignis des Layers Rails App Server ein Rezept zuzuweisen

1. Klicken Sie auf der Seite AWS OpsWorks Stacks Layers für Rails App Server auf Rezepte und dann auf Bearbeiten.
2. Geben Sie unter Custom Chef Recipes (Benutzerdefinierte Chef-Rezepte) den vollständig qualifizierten Rezeptnamen zum Bereitstellungsereignis an und klicken Sie auf +. Das Format eines vollständig qualifizierten Rezeptnamens lautet *`cookbookname`*`::`*`recipename`* , wobei *`recipename`* nicht die Erweiterung `.rb` enthält. In diesem Beispiel lautet der vollständig berechtigte Name `redis-config::generate`. Klicken Sie anschließend auf Save (Speichern), um die Konfiguration des Layers zu aktualisieren.



Schritt 5: Hinzufügen von JSON-Zugriffsinformationen zur Stack-Konfiguration

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Das Rezept `generate.rb` greift auf Stack-Konfigurations- und JSON-Bereitstellungsattribute zurück, die Host- und Port-Angaben des Redis-Servers enthalten. Diese Attribute sind zwar Teil des `[:deploy]` Standard-Namespaces, werden aber nicht automatisch von Stacks definiert. AWS OpsWorks Stattdessen definieren Sie die Attribute und deren Werte, indem Sie ein benutzerdefiniertes JSON-Objekt zum Stack hinzufügen. Das folgende Beispiel zeigt das benutzerdefinierte JSON-Objekt für dieses Beispiel.

Hinzufügen von Zugriffsinformation zur Stack-Konfiguration und JSON-Bereitstellung

1. Klicken Sie auf der Seite AWS OpsWorks Stacks Stack auf Stack-Einstellungen und dann auf Bearbeiten.
2. Fügen Sie im Abschnitt Configuration Management (Konfigurationsverwaltung) Zugriffsinformationen zum Feld Custom Chef JSON (Benutzerdefinierte JSON-Chef-Dateien) hinzu. Es sollte etwa wie im folgenden Beispiel aussehen, mit diesen Änderungen:

   - Ersetzen Sie `elasticache_redis_example` mit der Kurzbezeichnung Ihrer Anwendung.
   - Ersetzen Sie die `port` Werte `host` und durch die Werte für die ElastiCache Redis-Serverinstanz, in der Sie sie erstellt haben. Schritt 1: Erstellen Sie einen ElastiCache Redis-Cluster

```
{
  "deploy": {
     "elasticache_redis_example": {
       "redis": {
         "host": "mycluster.XXXXXXXXX.amazonaws.com",
         "port": "6379"
       }
     }
  }
}
```

Der Vorteil dieses Ansatzes besteht darin, dass Sie den Port- oder Host-Wert jederzeit ändern können, ohne Ihr benutzerdefiniertes Kochbuch zu berühren. AWS OpsWorks Stacks führt benutzerdefiniertes JSON mit dem integrierten JSON zusammen und installiert es auf den Instanzen des Stacks für alle nachfolgenden Lebenszyklusereignisse. Die Anwendungen können dann auf die Attributwerte mithilfe der Chef-Knotensyntax zugreifen, wie in Schritt 3: Erstellen und Bereitstellen eines benutzerdefinierten Rezeptbuchs beschrieben. Wenn Sie das nächste Mal eine Anwendung bereitstellen, installiert AWS OpsWorks Stacks eine Stack-Konfiguration und eine JSON-Bereitstellung, die die neuen Definitionen enthält, und `generate.rb` erstellt eine Konfigurationsdatei mit den aktualisierten Host- und Port-Angaben.

> **Note**
>
> `[:deploy]` fügt automatisch ein Attribut für jede bereitgestellte Anwendung hinzu, sodass `[:deploy][elasticache_redis_example]` bereits im Stack und in der JSON-Bereitstellung enthalten ist. `[:deploy][elasticache_redis_example]`Enthält jedoch kein `[:redis]` Attribut. Wenn Sie sie mit benutzerdefiniertem JSON definieren, wird AWS OpsWorks Stacks angewiesen, diese Attribute hinzuzufügen. `[:deploy][elasticache_redis_example]` Sie können auch ein benutzerdefiniertes JSON-Objekt verwenden, um vorhandene Attribute zu überschreiben. Weitere Informationen finden Sie unter Überschreiben der Attribute.

Schritt 6: Bereitstellen und Ausführen der Anwendung

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Beispiel wird davon ausgegangen, dass Sie über eine Ruby on Rails-Anwendung verfügen, die Redis verwendet. Für den Zugriff auf die Konfigurationsdatei können Sie Ihrer Gemfile-Datei das `redis`-Gem hinzufügen und eine Rails-Initialisierungsroutine in `config/initializers/redis.rb` folgendermaßen erstellen:

```
REDIS_CONFIG = YAML::load_file(Rails.root.join('config', 'redis.yml'))
$redis = Redis.new(:host => REDIS_CONFIG['host'], :port => REDIS_CONFIG['port'])
```

Erstellen Sie dann eine App, die Ihre Anwendung repräsentiert, und stellen Sie sie auf den Instanzen der Rails App Server-Ebene bereit. Dadurch wird der Anwendungscode aktualisiert und ausgeführt, `generate.rb` um die Konfigurationsdatei zu generieren. Wenn Sie die Anwendung ausführen, verwendet sie die ElastiCache Redis-Instanz als Schlüsselwertspeicher im Speicher.

Verwenden eines Amazon S3 S3-Buckets

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Anwendungen verwenden häufig einen Amazon Simple Storage Service (Amazon S3) -Bucket, um große Objekte wie Bilder oder andere Mediendateien zu speichern. Obwohl AWS OpsWorks Stacks keine integrierte Unterstützung für Amazon S3 bietet, können Sie einen Stack einfach so anpassen, dass Ihre Anwendung Amazon S3 S3-Speicher verwenden kann. In diesem Thema werden Sie anhand eines Linux-Stacks mit einem PHP-Anwendungsserver durch den grundlegenden Prozess der Bereitstellung von Amazon S3 S3-Zugriff auf Anwendungen geführt. Die grundlegenden Prinzipien gelten auch für Windows-Stacks.

Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Themen

- Schritt 5: Hinzufügen von Zugriffsinformation zu den Stack-Konfigurations- und JSON-Bereitstellungsattributen
- Schritt 6: Bereitstellen und Ausführen PhotoApp

Schritt 1: Erstellen Sie einen Amazon S3 S3-Bucket

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Sie müssen zuerst einen Amazon S3 S3-Bucket erstellen. Sie können dies direkt über die Amazon S3 S3-Konsole, API oder CLI tun. Eine einfachere Methode zum Erstellen von Ressourcen besteht jedoch häufig darin, eine AWS CloudFormation Vorlage zu verwenden. Die folgende Vorlage erstellt einen Amazon S3 S3-Bucket für dieses Beispiel und richtet ein Instance-Profil mit einer IAM-Rolle ein, die uneingeschränkten Zugriff auf den Bucket gewährt. Anschließend können Sie mit einer Layer-Einstellung das Instance-Profil den Anwendungsserver-Instances des Stacks hinzufügen, womit der Anwendung der Zugriff auf das Bucket ermöglicht wird, wie später beschrieben. Die Nützlichkeit von Instance-Profilen ist nicht auf Amazon S3 beschränkt. Sie sind auch für die Integration einer Vielzahl von AWS-Services wertvoll.

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

Wenn Sie die Vorlage starten, geschieht Folgendes:

- Die `AWS::S3::Bucket` Ressource erstellt einen Amazon S3 S3-Bucket.
- Die Ressource `AWS::IAM::InstanceProfile` erstellt ein Instance-Profil, das den Anwendungsserver-Instances zugewiesen wird.
- Die Ressource `AWS::IAM::Role` erstellt die Rolle des Instance-Profils.
- Die `AWS::IAM::Policy` Ressource legt die Berechtigungen der Rolle fest, um uneingeschränkten Zugriff auf Amazon S3 S3-Buckets zu ermöglichen.
- Der Abschnitt `Outputs` zeigt in der AWS CloudFormation -Konsole die Bucket- und Instance-Profil-Namen an, nachdem Sie die Vorlage gestartet haben.

  Diese Werte benötigen Sie, um den Stack und die Anwendung einzurichten.

Weitere Informationen zum Erstellen von AWS CloudFormation Vorlagen finden Sie unter Learn Template Basics.

Um den Amazon S3 S3-Bucket zu erstellen

1. Kopieren Sie die Beispielvorlage in eine Textdatei auf Ihrem System.

   In diesem Beispiel wird die Datei als `appserver.template` bezeichnet.
2. Öffnen Sie die AWS CloudFormation-Konsole und wählen Sie Create Stack (Stack erstellen) aus.
3. Geben Sie in das Feld Stack Name (Stack-Name) den Stack-Namen ein.

   In diesem Beispiel lautet der Name **`AppServer`**.
4. Wählen Sie Upload template file (Vorlagendatei hochladen) und Browse (Durchsuchen), wählen Sie die Datei `appserver.template` aus, die Sie in Schritt 1 erstellt haben, und wählen Sie Next Step (Nächster Schritt).
5. Wählen Sie auf der Seite Specify Parameters (Parameter angeben) I acknowledge that this template may create IAM resources (Ich nehme zur Kenntnis, dass diese Vorlage IAM-Ressourcen erstellen kann) aus und klicken Sie anschließend auf Next Step (Nächster Schritt) auf jeder Seite des Assistenten bis zum Ende. Wählen Sie Erstellen.
6. Wenn der AppServerStack den Status CREATE_COMPLETE erreicht hat, wählen Sie ihn aus und klicken Sie auf die Registerkarte Outputs.

   Sie müssen eventuell einige Male die Aktualisierungsfunktion wählen, um den Status zu aktualisieren.

7. Notieren Sie auf der Registerkarte Ausgaben die InstanceProfileNameWerte BucketNameund für die spätere Verwendung.

> **Note**
>
> AWS CloudFormation verwendet den Begriff Stapel, um sich auf die Sammlung von Ressourcen zu beziehen, die anhand einer Vorlage erstellt wurden. Er ist nicht dasselbe wie ein AWS OpsWorks Stacks-Stack.

Schritt 2: Erstellen Sie einen PHP-App-Server-Stack

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

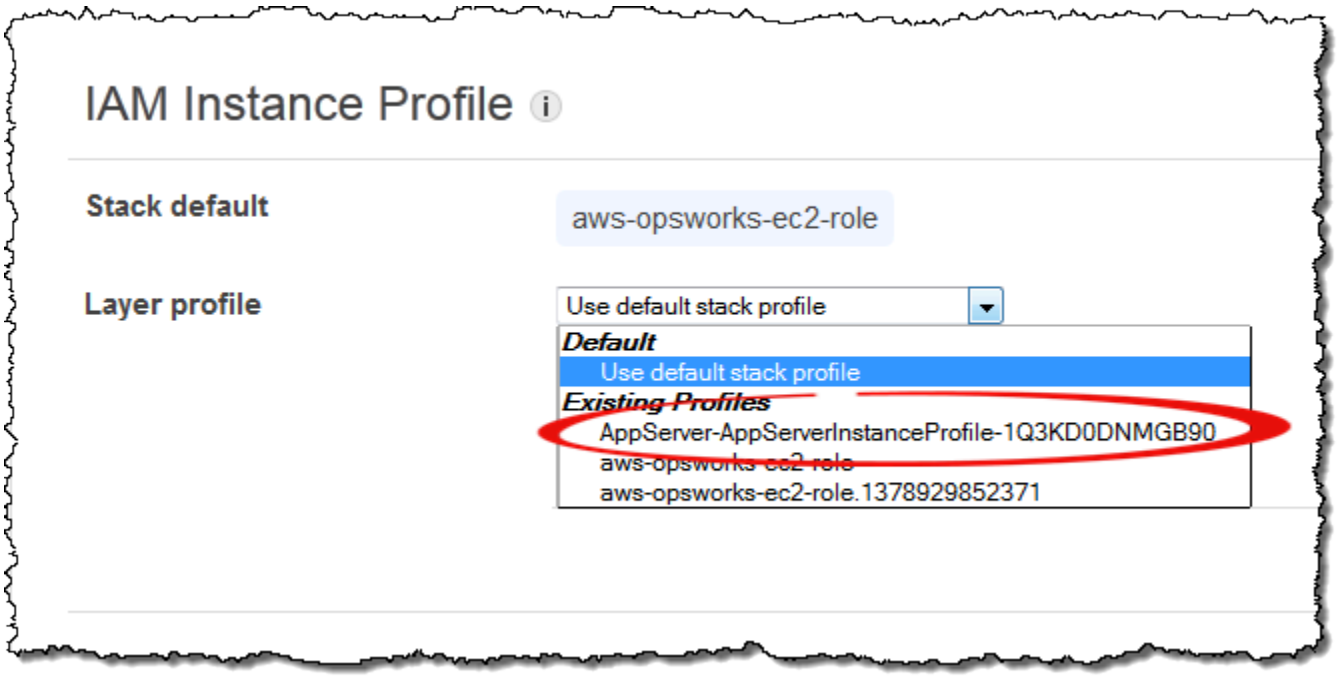
Der Stack besteht aus zwei Schichten, PHP App Server und MySQL, mit jeweils einer Instanz. Die Anwendung speichert Fotos in einem Amazon S3 S3-Bucket, verwendet jedoch die MySQL-Instance als Back-End-Datenspeicher, um Metadaten für jedes Foto zu speichern.

Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

So erstellen Sie den Stack

1. Erstellen Sie einen neuen Stack — benannt nach diesem **`PhotoSite`** Beispiel — und fügen Sie einen PHP-App-Server-Layer hinzu. Sie können für beide die Standardeinstellungen benutzen. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks und Eine OpsWorks Ebene erstellen .
2. Wählen Sie auf der Seite „Ebenen“ für PHP App Server die Option Sicherheit und dann Bearbeiten aus.

3. Wählen Sie im Abschnitt Layer-Profil den Namen des Instanzprofils aus, den Sie zuvor nach dem Start des AppServer AWS CloudFormation Stacks aufgezeichnet haben. Es wird so etwas wie sein`AppServer-AppServerInstanceProfile-1Q3KD0DNMGB90`. AWS OpsWorks Stacks weist dieses Profil allen Amazon EC2 EC2-Instances des Layers zu, wodurch Anwendungen, die auf den Instances des Layers ausgeführt werden, Zugriff auf Ihren Amazon S3 S3-Bucket erhalten.



4. Fügen Sie dem PHP App Server-Layer eine Instance hinzu und starten Sie sie. Weitere Informationen zum Hinzufügen und Starten von Instances finden Sie unter Hinzufügen einer Instance zu einem Layer.

5. Fügen Sie dem Stack eine MySQL-Ebene hinzu, fügen Sie eine Instanz hinzu und starten Sie sie. Sie können sowohl für den Layer als auch für die Instance die Standardeinstellungen verwenden. Insbesondere muss die MySQL-Instance nicht auf den Amazon S3 S3-Bucket zugreifen, sodass sie das standardmäßige AWS OpsWorks Stacks-Instance-Profil verwenden kann, das standardmäßig ausgewählt ist.

Schritt 3: Erstellen und Bereitstellen eines benutzerdefinierten Rezeptbuchs

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir

empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Der Stack ist noch nicht ganz bereit:

- Ihre Anwendung benötigt einige Informationen für den Zugriff auf den MySQL-Datenbankserver und den Amazon S3 S3-Bucket, z. B. den Datenbank-Hostnamen und den Amazon S3 S3-Bucket-Namen.
- Sie müssen eine Datenbank im MySQL-Datenbank-Server einrichten und eine Tabelle für die Metadaten der Fotos erstellen.

Sie könnten diese Aufgaben manuell erledigen, aber ein besserer Ansatz besteht darin, das Chef-Rezept zu implementieren und AWS OpsWorks Stacks das Rezept automatisch auf den entsprechenden Instances ausführen zu lassen. Chef-Rezepte sind spezialisierte Ruby-Anwendungen, mit denen AWS OpsWorks Stacks Aufgaben auf Instanzen wie das Installieren von Paketen oder das Erstellen von Konfigurationsdateien ausführt. Die Rezepte sind in einem Rezeptbuch gebündelt, das mehrere Rezepte und zugehörige Dateien enthalten kann, wie z. B. Vorlagen für Konfigurationsdateien. Das Kochbuch befindet sich in einem Repository wie GitHub und muss eine Standardverzeichnisstruktur haben. Wenn Sie noch nicht über ein benutzerdefiniertes Rezeptbuch-Repository verfügen, finden Sie unter Rezeptbuch-Repositorys Informationen, wie Sie es erstellen können.

In diesem Beispiel wurde das Kochbuch für Sie implementiert und in einem öffentlichen GitHub Repository gespeichert. Das Rezeptbuch enthält zwei Rezepte, `appsetup.rb` und `dbsetup.rb`, sowie eine Vorlagendatei, `db-connect.php.erb`.

Das `appsetup.rb` Rezept erstellt eine Konfigurationsdatei, die die Informationen enthält, die die Anwendung für den Zugriff auf die Datenbank und den Amazon S3 S3-Bucket benötigt. Im Grunde handelt es sich um eine leicht modifizierte Version des `appsetup.rb` Rezepts, welches in Verknüpfen der Anwendung mit der Datenbank beschrieben ist. Der primäre Unterschied besteht in den an die Vorlage übermittelten Variablen bezüglich den Zugriffsinformationen.

Die ersten vier Attribute definieren die Datenbankverbindungseinstellungen und werden automatisch von AWS OpsWorks Stacks definiert, wenn Sie die MySQL-Instanz erstellen.

Es gibt zwei Unterschiede zwischen diesen Variablen und jenen des Originalrezepts:

- Wie im Originalrezept stellt die Variable `table` den Namen der Datenbanktabelle dar, die von `dbsetup.rb` erstellt wird, und wird auf den Wert eines Attributs gesetzt, das in der Attribute-Datei des Rezeptbuchs definiert ist.

  Das Attribut hat jedoch einen anderen Namen: `[:photoapp][:dbtable]`.

- Die `s3bucket` Variable ist spezifisch für dieses Beispiel und wird auf den Wert eines Attributs gesetzt, das den Amazon S3 S3-Bucket-Namen darstellt,`[:photobucket]`.

  `[:photobucket]` wird unter Verwendung eines benutzerdefinierten JSON-Objekts definiert, wie später beschrieben. Weitere Informationen zu Attributen finden Sie unter Attribute.

Weitere Informationen zu Attributen finden Sie unter Attribute.

Das Rezept `dbsetup.rb` richtet eine Datenbanktabelle für die jeweiligen Foto-Metadaten ein. Im Grunde handelt es sich um eine leicht modifizierte Version des Rezepts `dbsetup.rb`, welches in Einrichten der Datenbank beschrieben ist. Dort finden Sie eine detaillierte Beschreibung.

Der einzige Unterschied zwischen diesem Beispiel und dem Originalrezept ist das Datenbankschema, das aus drei Spalten besteht, die die ID, URL und Bildunterschrift jedes Fotos enthalten, das im Amazon S3 S3-Bucket gespeichert ist.

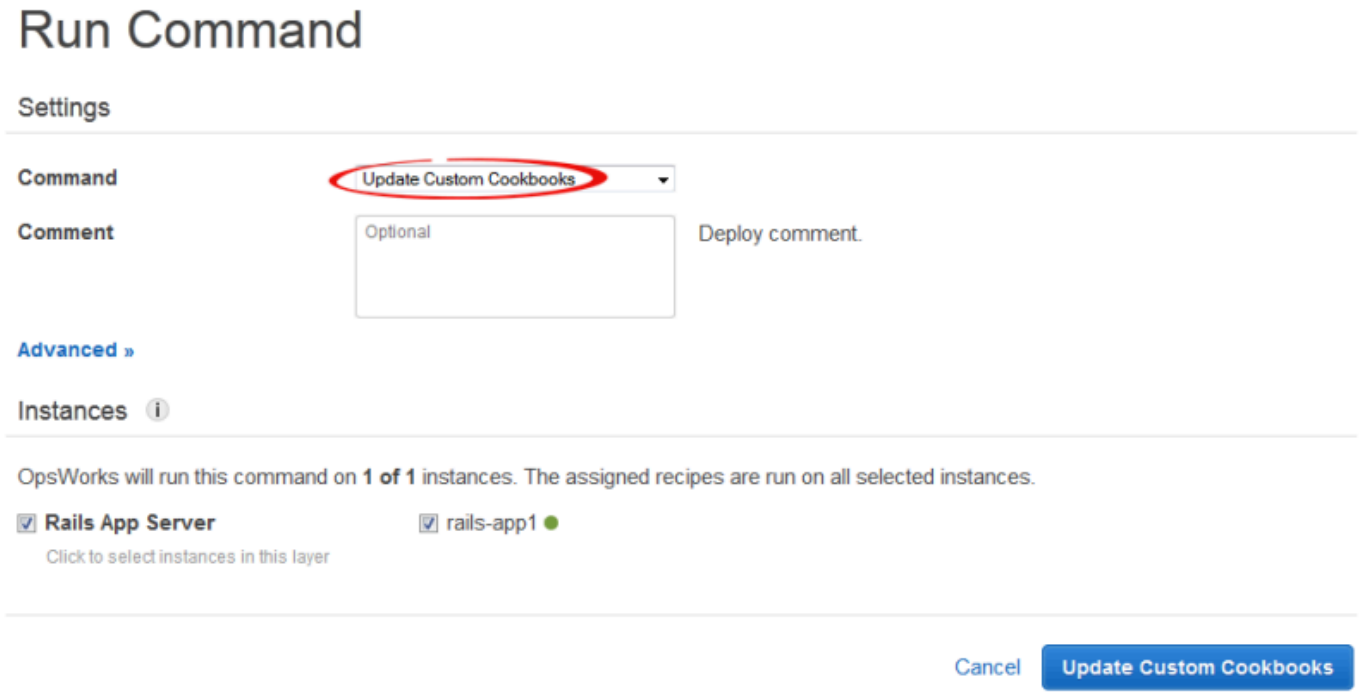
Die Rezepte sind bereits implementiert, sodass Sie lediglich das Photoapp-Kochbuch im Kochbuch-Cache jeder Instanz bereitstellen müssen. AWS OpsWorks Stacks führt dann die zwischengespeicherten Rezepte aus, wenn das entsprechende Lebenszyklusereignis eintritt, wie später beschrieben.

So stellen Sie ein PhotoApp-Rezeptbuch bereit

1. Wählen Sie auf der Seite AWS OpsWorks Stacks Stack die Option Stack-Einstellungen und dann Bearbeiten aus.
2. Im Abschnitt Configuration Management (Konfigurationsverwaltung):

   - Legen Sie Use custom Chef Cookbooks (Benutzerdefinierte Chef-Rezeptbücher verwenden) auf Yes (Ja) fest,
   - Legen Sie Repository type (Repository-Typ) auf "Git" fest.
   - Legen Sie Repository URL (Repository-URL) auf **`git://github.com/amazonwebservices/opsworks-example-cookbooks.git`** fest.

3. Wählen Sie auf der Seite Stack die Option Run Command (Befehl ausführen), wählen Sie den Stack-Befehl Update Custom Cookbooks (Benutzerdefinierte Kochbücher aktualisieren) aus und wählen dann Update Custom Cookbooks (Benutzerdefinierte Kochbücher aktualisieren), um das neue Rezeptbuch im Rezeptbuch-Cache der Instance zu installieren.



Schritt 4: Ordnen Sie die Rezepte Ereignissen zu LifeCycle

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

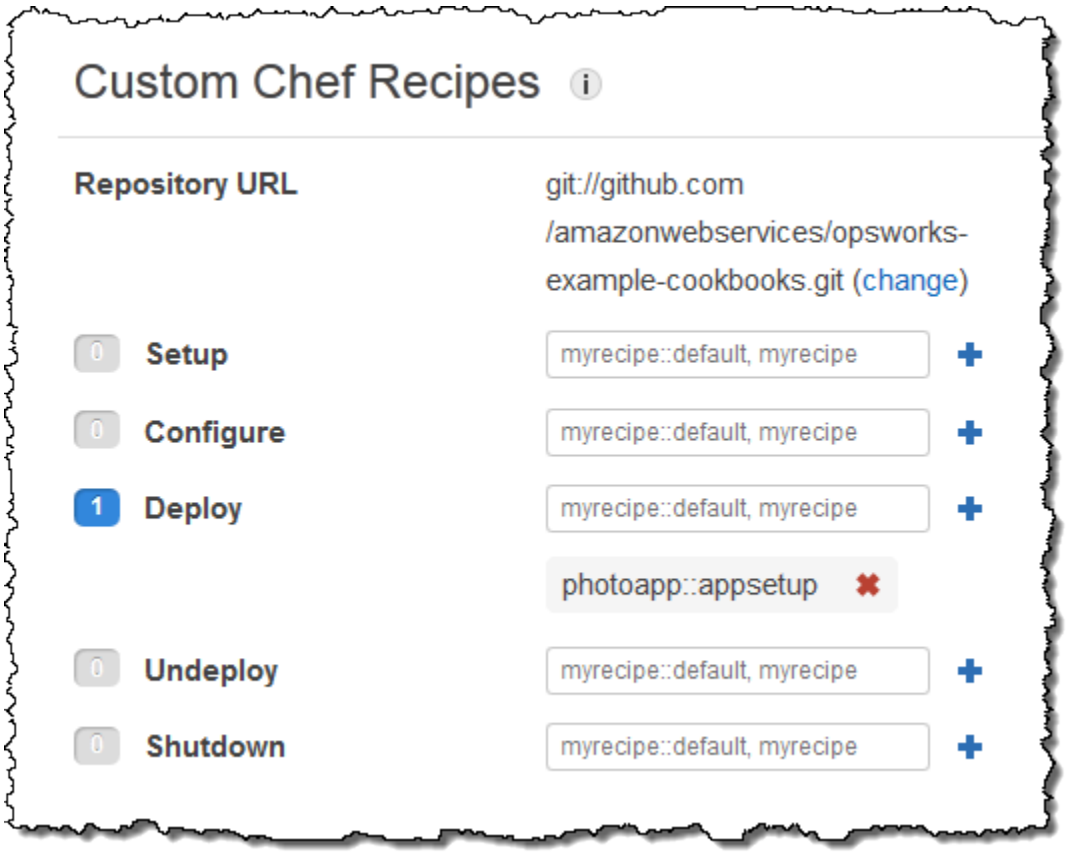
Du kannst benutzerdefinierte Rezepte manuell ausführen, aber der beste Ansatz ist normalerweise, sie von AWS OpsWorks Stacks automatisch ausführen zu lassen. Jeder Layer verfügt über einen Satz integrierter Rezepte, die jedem der fünf Lebenszyklusereignisse — Setup, Configure, Deploy, Undeploy und Shutdown — zugewiesen sind. Jedes Mal, wenn ein Ereignis auf einer Instanz eintritt, führt AWS OpsWorks Stacks die zugehörigen Rezepte für jede Ebene der Instanz aus, die die

erforderlichen Aufgaben erledigen. Wenn eine Instanz beispielsweise den Startvorgang beendet hat, löst AWS OpsWorks Stacks ein Setup-Ereignis aus, um die Setup-Rezepte auszuführen, die normalerweise Aufgaben wie das Installieren und Konfigurieren von Paketen übernehmen.

Sie können AWS OpsWorks Stacks benutzerdefinierte Rezepte für die Instanzen einer Ebene ausführen lassen, indem Sie jedes Rezept dem entsprechenden Lebenszyklusereignis zuweisen. AWS OpsWorks Stacks führt alle benutzerdefinierten Rezepte aus, nachdem die integrierten Rezepte der Ebene abgeschlossen sind. In diesem Beispiel weisen Sie `appsetup.rb` dem Deploy-Ereignis der PHP App Server-Ebene und dem Deploy-Ereignis der MySQL-Schicht `dbsetup.rb` zu. AWS OpsWorks Stacks führt die Rezepte dann beim Start, nach Abschluss der integrierten Setup-Rezepte und jedes Mal, wenn Sie eine App bereitstellen, nachdem die erstellten Deploy-Rezepte abgeschlossen sind, auf den Instanzen der zugehörigen Ebene aus. Weitere Informationen finden Sie unter Automatisches Ausführen von Rezepten.

So weisen Sie benutzerdefinierte Rezepte zum Bereitstellungsereignis des Layers zu

1. Wählen Sie auf der Seite AWS OpsWorks Stacks Layers für den PHP App Server die Option Rezepte und dann Bearbeiten aus.
2. Geben Sie unter Custom Chef Recipes (Benutzerdefinierte Chef-Rezepte) den vollständig berechtigten Rezeptnamen zum Bereitstellungsereignis an und wählen Sie +. Der Name muss dem Chef-Format *`cookbookname`*`::`*`recipename`* entsprechen, wobei *`recipename`* ohne die Erweiterung `.rb` angegeben wird. In diesem Beispiel geben Sie `photoapp::appsetup` ein. Wählen Sie anschließend Save (Speichern), um die Konfiguration des Layers zu aktualisieren.



3. Wählen Sie auf der Seite Ebenen in der Spalte Aktionen der MySQL-Ebene die Option Bearbeiten aus.
4. Fügen Sie `photoapp::dbsetup` zum Bereitstellungsereignis des Layers hinzu und speichern Sie die neue Konfiguration.

Schritt 5: Hinzufügen von Zugriffsinformation zu den Stack-Konfigurations- und JSON-Bereitstellungsattributen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Das `appsetup.rb` Rezept hängt von Daten aus der AWS OpsWorks Stacks-Stack-Konfiguration und den Bereitstellungsattributen ab, die auf jeder Instanz installiert sind und detaillierte Informationen über den Stack und alle bereitgestellten Apps enthalten. Die `deploy`-Attribute des Objekts haben folgende Struktur, die der Einfachheit halber als JSON angezeigt wird:

```
{
   ...
  "deploy": {
    "app1": {
      "application" : "short_name",
      ...
    }
    "app2": {
      ...
    }
    ...
  }
}
```

Der Bereitstellungsknoten enthält ein Attribut für jede bereitgestellte Anwendung, die mit dem Kurznamen der Anwendung bezeichnet wird. Jedes Anwendungsattribut enthält eine Gruppe von Attributen, die die Konfiguration der Anwendung definieren, wie beispielsweise das Dokument-Stammverzeichnis und den Anwendungstyp. Eine Liste der `deploy`-Attribute finden Sie unter Bereitstellungsattribute. Sie können die Werte der Stack-Konfigurations- und Bereitstellungsattribute in Ihren Rezepten unter Verwendung der Chef-Attributsyntax wiedergeben. Beispielsweise stellt `[:deploy][:app1][:application]` den Kurznamen der Anwendung App1 dar.

Die benutzerdefinierten Rezepte hängen von verschiedenen Stackkonfigurations- und Bereitstellungsattributen ab, die Datenbank- und Amazon S3 S3-Zugriffsinformationen darstellen:

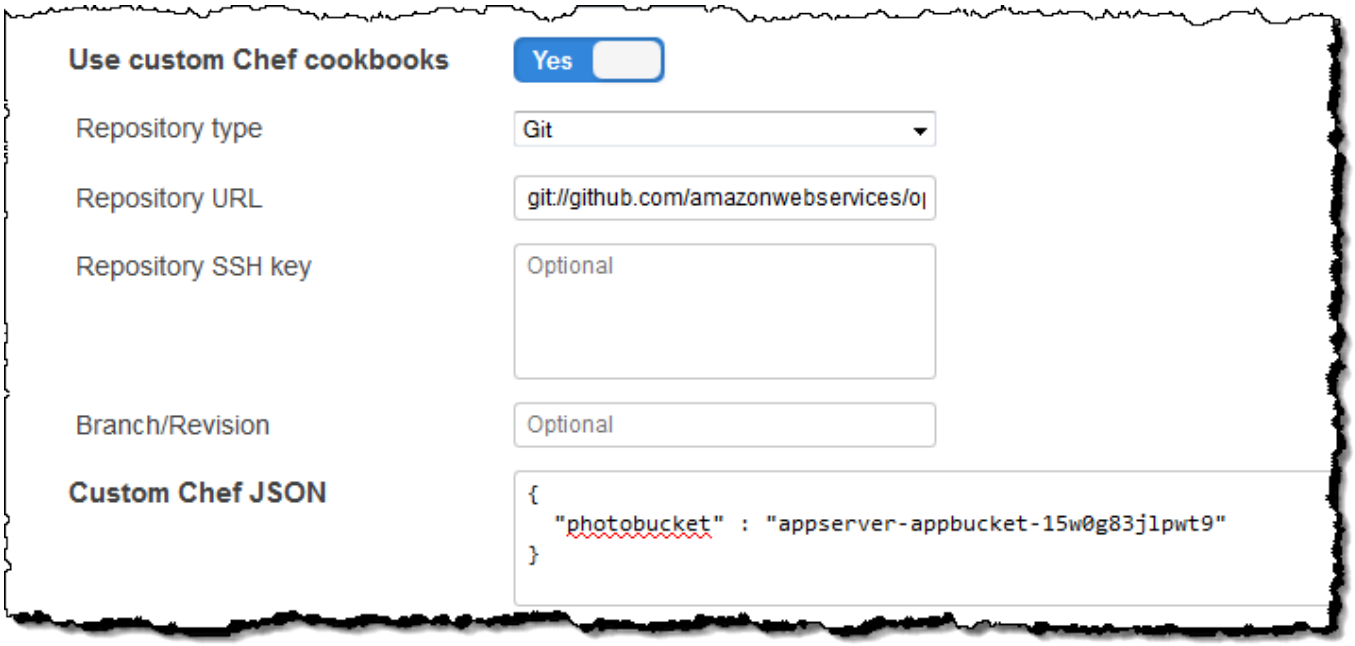
- Die Datenbankverbindungsattribute, z. B.`[:deploy][:database][:host]`, werden von AWS OpsWorks Stacks definiert, wenn es die MySQL-Schicht erstellt.
- Das Attribut für den Tabellennamen `[:photoapp][:dbtable]` wird in der Attributdatei im benutzerdefinierten Rezeptbuch definiert und ist auf `foto` gesetzt.
- Sie müssen das Attribut für den Bucket-Namen definieren, `[:photobucket]`, indem Sie mithilfe des benutzerdefinierten JSON-Objekts das Attribut zu den Stack-Konfigurations- und Bereitstellungsattributen hinzufügen.

So definieren Sie das Amazon S3 S3-Bucket-Name-Attribut

1. Wählen Sie auf der Seite AWS OpsWorks Stacks Stack die Option Stack-Einstellungen und dann Bearbeiten aus.
2. Fügen Sie im Abschnitt Configuration Management (Konfigurationsverwaltung) Zugriffsinformationen zum Feld Custom Chef JSON (Benutzerdefinierte JSON-Chef-Dateien) hinzu. Es sollte etwa wie folgt aussehen:

```
{
  "photobucket" : "yourbucketname"
}
```

   Ersetzen Sie *IhrBucketName* mit dem Bucket-Namen, den Sie in Schritt 1: Erstellen Sie einen Amazon S3 S3-Bucket notiert haben.



AWS OpsWorks Stacks führt das benutzerdefinierte JSON mit den Stackkonfigurations- und Bereitstellungsattributen zusammen, bevor es sie auf den Instanzen des Stacks installiert. Anschließend `appsetup.rb` kann der Bucket-Name aus dem Attribut abgerufen werden. `[:photobucket]` Wenn Sie den Bucket ändern möchten, müssen Sie nicht das Rezept bearbeiten. Sie können einfach das Attribut überschreiben, um einen neuen Bucket-Namen festzulegen.

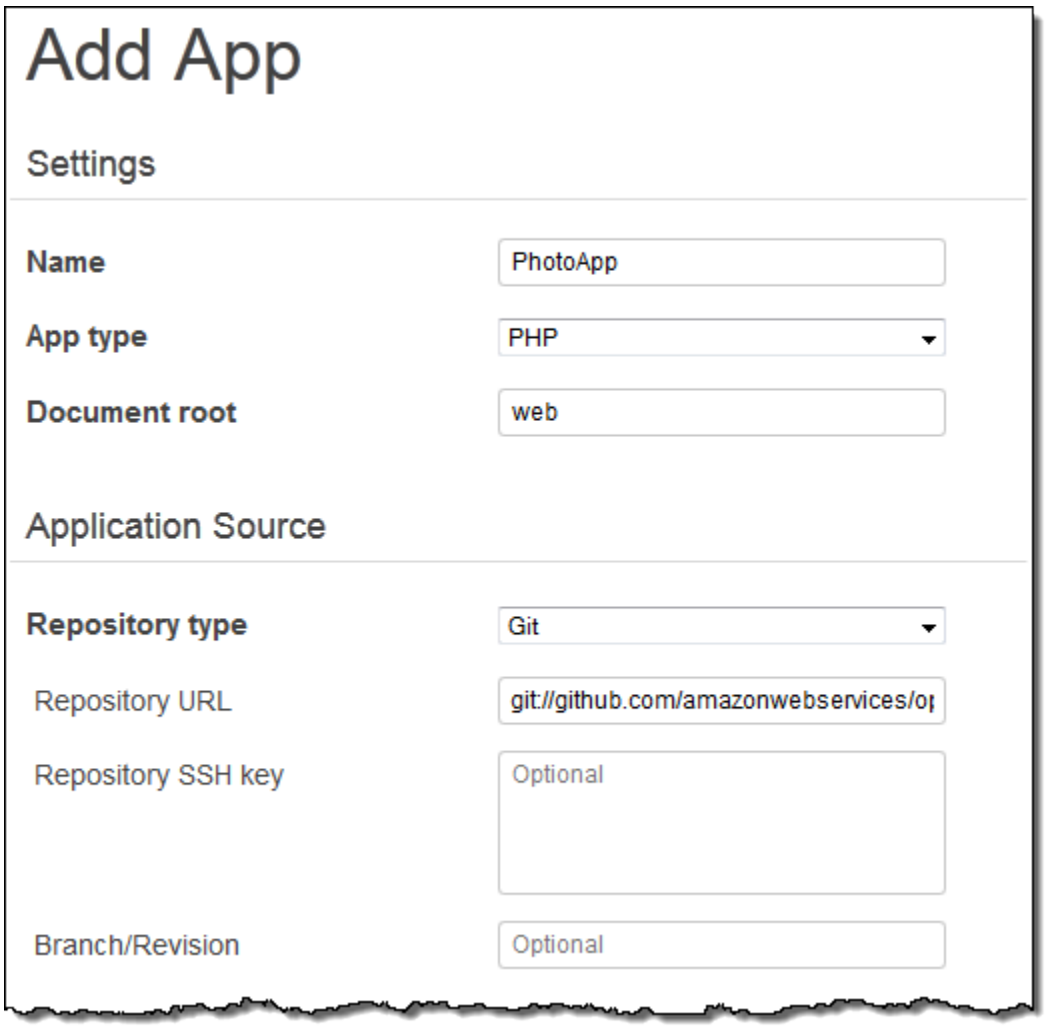
Schritt 6: Bereitstellen und Ausführen PhotoApp

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

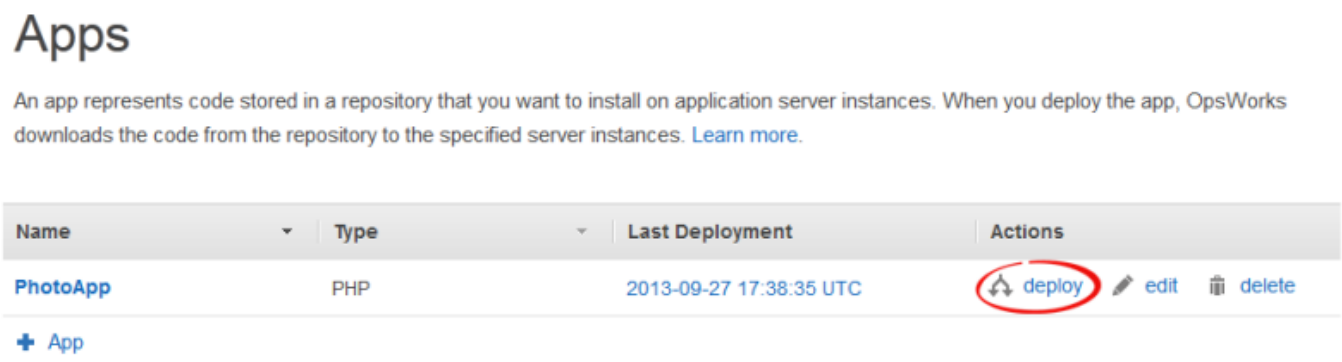
In diesem Beispiel wurde die Anwendung ebenfalls für Sie implementiert und ist in einem öffentlichen GitHub Repository gespeichert. Sie müssen lediglich die Anwendung zum Stack hinzufügen, für die Anwendungsserver bereitstellen und ausführen.

So fügen Sie die Anwendung zum Stack hinzu und stellen diese für die Anwendungsserver bereit

1. Öffnen Sie die Seite Apps und wählen Sie Add an app (Eine App hinzufügen) aus.
2. Führen Sie auf der Seite Add App (App hinzufügen) die folgenden Schritte aus:
   - Legen Sie Name auf **`PhotoApp`** fest.
   - Legen Sie App type (App-Typ) auf PHP fest.
   - Legen Sie Document root (Dokumentenstamm) auf **`web`** fest.
   - Legen Sie Repository type (Repository-Typ) auf Git fest.
   - Legen Sie Repository URL (Repository-URL) auf **`git://github.com/awslabs/opsworks-demo-php-photo-share-app.git`** fest.
   - Wählen Sie Add App (App hinzufügen), um für die restlichen Einstellungen die Standardwerte zu übernehmen.

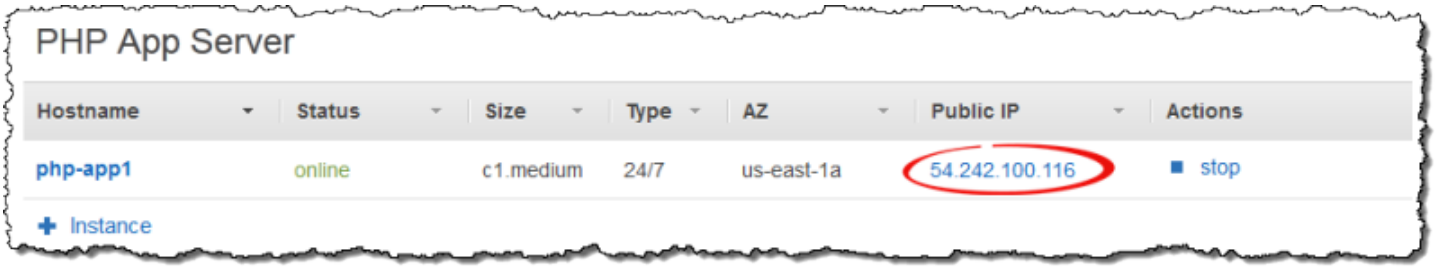


3. Wählen Sie auf der Seite Apps in der Spalte Aktionen der PhotoApp App die Option Bereitstellen aus.



4. Akzeptieren Sie die Standardwerte und wählen Sie Deploy (Bereitstellen), um die Anwendung für den Server bereitzustellen.

Gehen Sie zur Ausführung PhotoApp auf die Seite „Instanzen“ und wählen Sie die öffentliche IP-Adresse der PHP App Server-Instanz aus.



Folgende Benutzeroberfläche sollte angezeigt werden. Wählen Sie Foto hinzufügen, um ein Foto im Amazon S3 S3-Bucket und die Metadaten im Back-End-Datenspeicher zu speichern.

AWS CodePipeline Mit AWS OpsWorks Stacks verwenden

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS CodePipelineermöglicht es Ihnen, Continuous-Delivery-Pipelines zu erstellen CodeCommit, die Codeänderungen aus Quellen wie Amazon Simple Storage Service (Amazon S3) oder GitHubverfolgen. Sie können CodePipeline damit die Veröffentlichung Ihrer Chef-Kochbücher und des Anwendungscodes für AWS OpsWorks Stacks auf Chef 11.10-, Chef 12- und Chef 12.2-Stacks automatisieren. Beispiele in diesem Abschnitt beschreiben, wie Sie eine einfache Pipeline CodePipeline als Bereitstellungstool für Code erstellen und verwenden, den Sie auf Stacks-Layern ausführen. AWS OpsWorks

> **Note**
>
> CodePipeline und die AWS OpsWorks Stacks-Integration wird für die Bereitstellung auf Chef 11.4 und älteren Stacks nicht unterstützt.

Themen

AWS CodePipeline mit AWS OpsWorks Stacks - Chef 12 Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS CodePipelineermöglicht es Ihnen, Continuous-Delivery-Pipelines zu erstellen CodeCommit, die Codeänderungen aus Quellen wie Amazon Simple Storage Service (Amazon S3) oder GitHubverfolgen. Das Beispiel in diesem Thema beschreibt, wie Sie eine einfache Pipeline CodePipeline als Bereitstellungstool für Code erstellen und verwenden, den Sie auf AWS OpsWorks Stacks-Layern ausführen. In diesem Beispiel erstellen Sie eine Pipeline für eine einfache App Node.js und weisen AWS OpsWorks Stacks dann an, die App auf allen Instanzen in einer Ebene in einem Chef 12-Stapel (in diesem Fall einer einzelnen Instanz) auszuführen.

> **Note**
>
> In diesem Thema wird beschrieben, wie Sie eine Pipeline für die Ausführung und Aktualisierung einer App auf einem Chef 12-Stack verwenden. Weitere Informationen dazu, wie Sie mithilfe einer Pipeline Apps auf einem Chef 11.10-Stack ausführen und aktualisieren, finden Sie unter AWS CodePipeline mit AWS OpsWorks Stacks - Chef 11 Stacks. Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere

Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Themen

Voraussetzungen

Szellen Sie sicher, dass Sie über Administratorberechtigungen für die folgenden Aufgaben verfügen, bevor Sie diese Anleitung starten. Sie können Mitglied einer Gruppe sein, auf die die AdministratorAccessRichtlinie angewendet wurde, oder Sie können Mitglied einer Gruppe sein, die über die in der folgenden Tabelle aufgeführten Berechtigungen und Richtlinien verfügt. Aus Sicherheitsgründen sollten Sie einer Gruppe angehören, die über die erforderlichen Rechte für die folgenden Aufgaben verfügt, anstatt einzelnen Benutzern die erforderlichen Berechtigungen zuzuweisen.

Weitere Informationen zum Erstellen einer Sicherheitsgruppe in IAM und zum Zuweisen von Berechtigungen zu dieser Gruppe finden Sie unter IAM-Benutzergruppen erstellen. Weitere Informationen zur Verwaltung von AWS OpsWorks Stacks-Berechtigungen finden Sie unter Bewährte Methoden: Berechtigungen verwalten.

Berechtigungen	Empfohlene Richtlinie für das Anfügen an eine Gruppe
Erstellen und bearbeiten Sie Stapel, Ebenen und Instanzen in AWS OpsWorks Stacks.	AWSOpsWorks_FullAccess
Erstellen, bearbeiten und führen Sie AWS CloudFormation-Vorlagen aus.	AmazonCloudFormationFullAccess
Erstellen, bearbeiten und greifen Sie auf Amazon S3 S3-Buckets zu.	Amazon S3 FullAccess
Pipelines erstellen, bearbeiten und ausführen , insbesondere in Pipelines CodePipeline, die AWS OpsWorks Stacks als Anbieter verwenden.	AWSCodePipeline_FullAccess

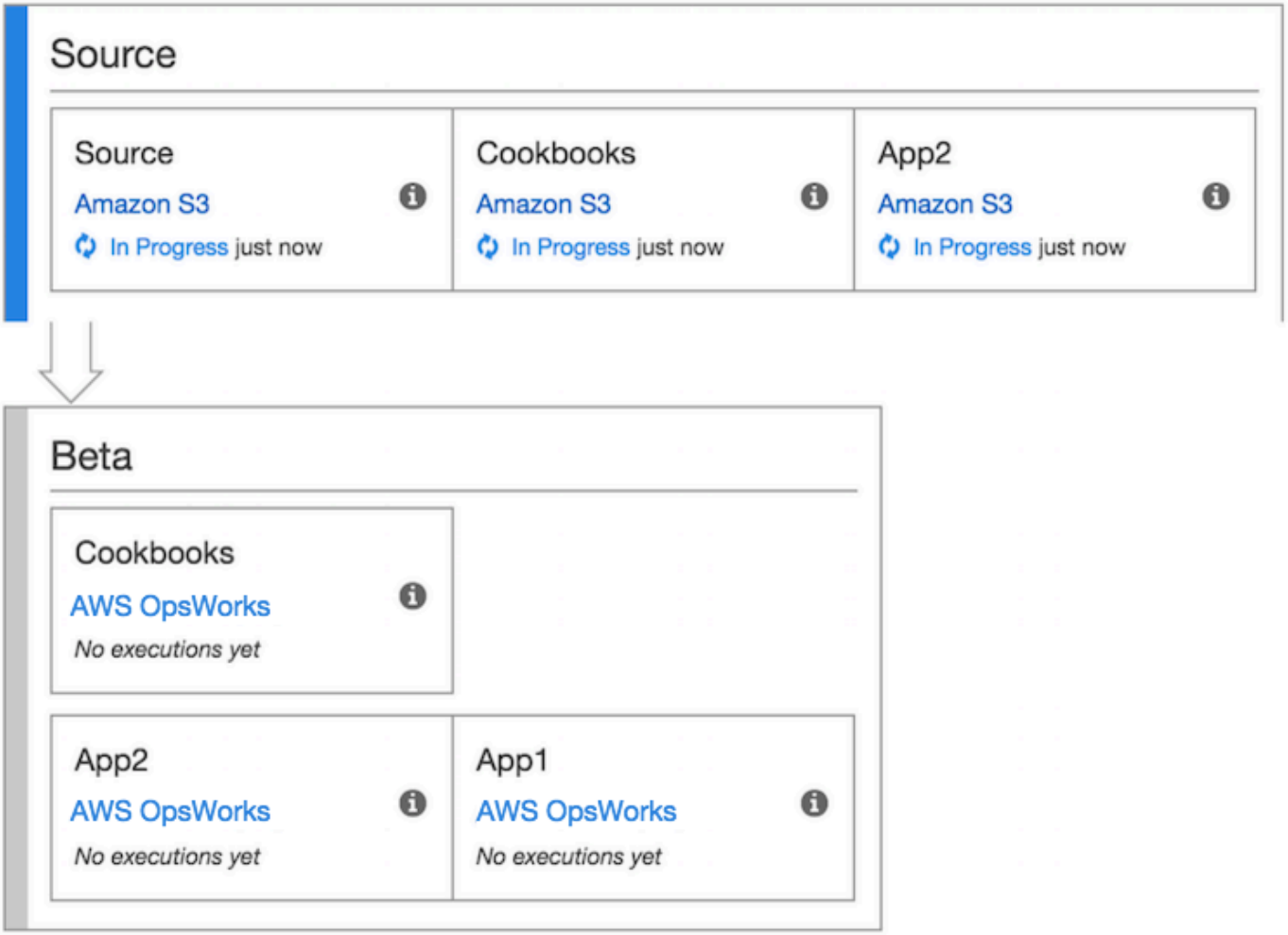
Sie benötigen außerdem ein Amazon EC2 EC2-Schlüsselpaar. Sie werden aufgefordert, den Namen dieses key pair anzugeben, wenn Sie die AWS CloudFormation Vorlage ausführen, mit der der Beispielstapel, die Ebene und die Instanz in dieser exemplarischen Vorgehensweise erstellt werden. Weitere Informationen zum Abrufen eines key pair in der Amazon EC2 EC2-Konsole finden Sie unter Create a Key Pair in der Amazon EC2 EC2-Dokumentation. Das key pair muss sich in der Region USA Ost (Nord-Virginia) befinden. Sie können ein vorhandenes Schlüsselpaar verwenden, wenn Sie in der betreffenden Region bereits über ein Schlüsselpaar verfügen.

Andere unterstützte Szenarien

Diese Anleitung erstellt eine einfache Pipeline, die die Stufen Source (Quelle) und Deploy (Bereitstellen) umfasst. Sie können jedoch komplexere Pipelines erstellen, die AWS OpsWorks Stacks als Anbieter verwenden. Im Folgenden werden einige Beispiele für unterstützte Pipelines und Szenarien aufgeführt:

- Sie können eine Pipeline bearbeiten, um ein Chef-Rezeptbuch der Stufe Source (Quelle) und ein zugehöriges Ziel für aktualisierte Rezeptbücher der Stufe Deploy (Bereitstellen) hinzuzufügen. In diesem Fall fügen Sie eine Deploy (Bereitstellen)-Aktion hinzu, die eine Aktualisierung Ihrer Rezeptbücher auslöst, wenn Sie Änderungen an der Quelle vornehmen. Das aktualisierte Rezeptbuch wird vor Ihrer Anwendung bereitgestellt.

- Sie können eine komplexe Pipeline mit benutzerdefinierten Kochbüchern und mehreren Apps erstellen und diese in einem AWS OpsWorks Stacks-Stack bereitstellen. Die Pipeline verfolgt Änderungen an der Anwendung und den Rezeptbuchquellen und stellt sich erneut bereit, wenn Sie Änderungen vorgenommen haben. Die folgende Abbildung zeigt ein Beispiel einer ähnlichen, komplexen Pipeline:



Weitere Informationen zur Arbeit mit CodePipeline finden Sie im CodePipeline Benutzerhandbuch.

Schritt 1: Erstellen Sie einen Stapel, eine Ebene und eine Instanz in AWS OpsWorks Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir

empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Um AWS OpsWorks Stacks als Bereitstellungsanbieter für eine Pipeline verwenden zu können, müssen Sie zunächst über einen Stack, eine Ebene und mindestens eine Instanz in der Ebene verfügen. Sie können zwar einen Stack in AWS OpsWorks Stacks erstellen, indem Sie den Anweisungen unter Erste Schritte mit Linux Stacks oder Erste Schritte mit Windows Stacks folgen. Um Zeit zu sparen, verwendet dieses Beispiel jedoch eine AWS CloudFormation Vorlage, um einen Linux-basierten Chef 12-Stack, -Layer und -Instanz zu erstellen. Die Instance, die durch diese Vorlage erstellt wurde, führt Amazon Linux 2016.03 aus und hat den Instance-Typ `c3.large`. Ihr Stack wird über die Vorlage nicht zur Verwendung von benutzerdefinierten Rezeptbüchern konfiguriert – dies werden wir im Rahmen dieser Anleitung manuell tun.

Important

Die AWS CloudFormation Vorlage muss in derselben Region gespeichert und ausgeführt werden wie der Amazon S3 S3-Bucket, in den Sie später Ihre App hochladen, und in derselben Region, in der Sie später Ihre Pipeline erstellen CodePipeline. CodePipeline Unterstützt derzeit nur den AWS OpsWorks Stacks-Anbieter in der Region USA Ost (Nord-Virginia) (us-east-1). Alle Ressourcen in dieser exemplarischen Vorgehensweise sollten in der Region USA Ost (Nord-Virginia) erstellt werden.
Wenn das Erstellen des Stacks misslingt, haben Sie möglicherweise die maximal zulässige Anzahl der IAM-Rollen für Ihr Konto fast erreicht. Die Stack-Erstellung kann auch fehlschlagen, wenn Ihr Konto Instances des Instance-Typen `c3.large` nicht starten kann. Wenn Sie beispielsweise das AWS kostenlose Kontingent verwenden, erhalten Sie möglicherweise eine Fehlermeldung wie`Root device type: must be included in EBS`. Wenn Ihr Konto Einschränkungen in Bezug auf die Instance-Typen hat, die Sie erstellen dürfen, z. B. Einschränkungen, die durch das AWS kostenlose Kontingent auferlegt werden, versuchen Sie, den Wert des `InstanceType` Parameters im Instance-Block der Vorlage auf einen Instance-Typ zu ändern, den Ihr Konto verwenden kann.

Um einen Stack, eine Ebene und eine Instanz zu erstellen, verwenden Sie AWS CloudFormation

1. Kopieren Sie die folgende AWS CloudFormation Vorlage in ein neues Klartext-Dokument. Speichern Sie die Datei an einem geeigneten Ort auf Ihrem lokalen Computer und geben Sie ihr den Namen NewOpsWorksStack.template oder einen anderen für Sie passenden Namen.

```
{
  "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",
```

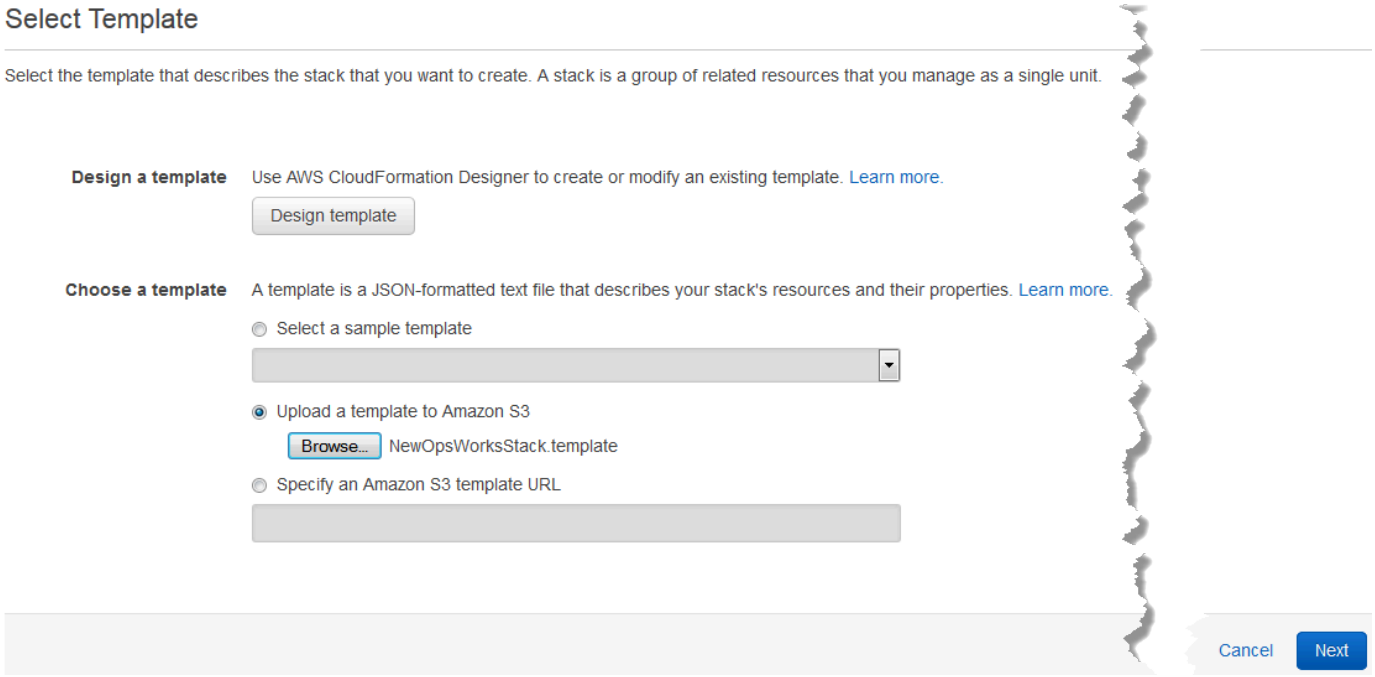
```
"EnableAutoHealing": "true",
      "AutoAssignElasticIps": "false",
      "AutoAssignPublicIps": "true",
"CustomSecurityGroupIds": [
  {
    "Fn::GetAtt": [
           "CPOpsDeploySecGroup", "GroupId"
    ]
  }
 ]
    },
    "DependsOn": [
      "MyStack",
      "CPOpsDeploySecGroup"
    ]
  },
  "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. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie auf der AWS CloudFormation Startseite die Option Stack erstellen aus.
4. Wählen Sie auf der Seite Select Template (Vorlage auswählen) im Bereich Choose a template (Eine Vorlage auswählen) die Option Upload a template to Amazon S3 (Eine Vorlage in Amazon S3 hochladen) und anschließend Browse (Durchsuchen) aus.
5. Navigieren Sie zu der AWS CloudFormation Vorlage, die Sie in Schritt 1 gespeichert haben, und wählen Sie dann Öffnen aus. Wählen Sie auf der Seite Vorlage auswählen die Option Weiter aus.



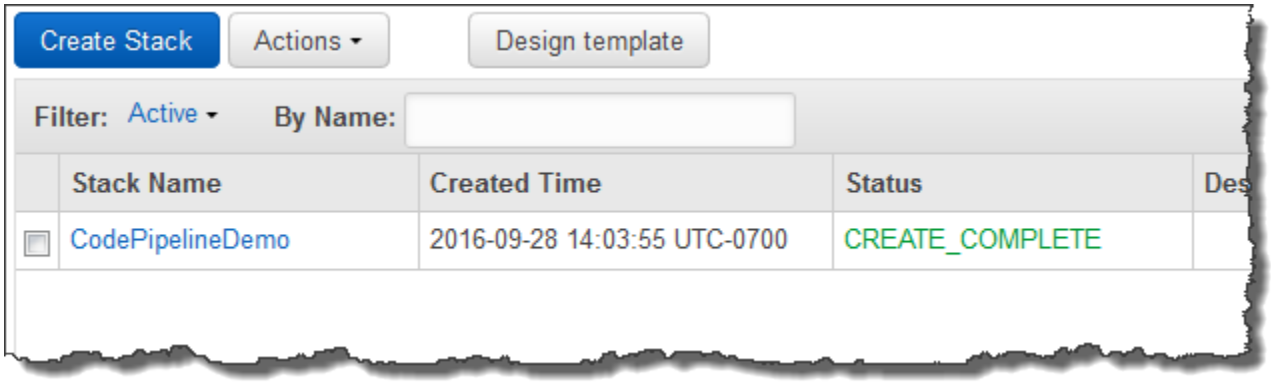
6. Geben Sie auf der Seite „Details angeben“ den Stack oder einen beliebigen Stacknamen CodePipelineDemo, der für Ihr Konto eindeutig ist, einen Namen. Wenn Sie einen anderen Namen für Ihren Stack auswählen, ändern Sie den Stack-Namen beim Ausführen dieser Anleitung.
7. Geben Sie im Bereich Parameter den Namen eines EC2-Schlüsselpaars ein, das Sie für den Zugriff auf Ihre AWS OpsWorks Stacks-Instance verwenden möchten, nachdem sie erstellt wurde. Wählen Sie Weiter aus.
8. Wählen Sie auf der Seite Optionen Weiter aus. (Einstellungen auf dieser Seite sind für diese Anleitung nicht erforderlich.)
9. Die AWS CloudFormation Vorlage, die Sie in dieser exemplarischen Vorgehensweise verwenden, erstellt IAM-Rollen, ein Instanzprofil und eine Instanz.

> **Important**
>
> Bevor Sie „Erstellen“ wählen, wählen Sie „Kosten“ aus, um die Kosten zu schätzen, die Ihnen durch die Erstellung von AWS Ressourcen mit dieser Vorlage entstehen könnten.

Wenn das Erstellen von IAM-Ressourcen zulässig ist, aktivieren Sie das Kontrollkästchen Ich bestätige, dass diese Vorlage möglicherweise AWS CloudFormation dazu führt, dass IAM-

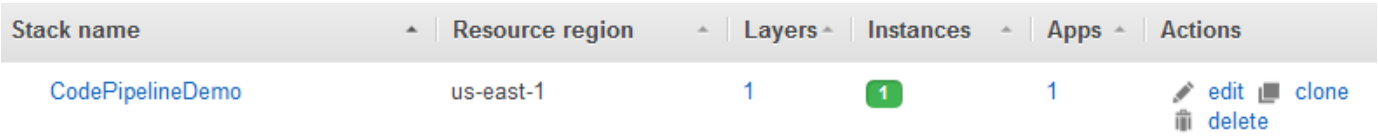
Ressourcen erstellt werden, und wählen Sie dann Erstellen aus. Wenn das Erstellen von IAM-Ressourcen nicht zulässig ist, können Sie mit diesem Verfahren nicht fortfahren.

10. Auf dem AWS CloudFormation Dashboard können Sie den Fortschritt der Erstellung des Stacks verfolgen. Bevor Sie mit dem nächsten Schritt fortfahren, warten Sie, bis CREATE_COMPLETE in der Spalte Status angezeigt wird.



Um die Stack-Erstellung in AWS OpsWorks Stacks zu überprüfen

1. Öffnen Sie die AWS OpsWorks Konsole unter https://console.aws.amazon.com/opsworks/.
2. Sehen Sie sich im AWS OpsWorks Stacks-Dashboard den Stack an, den Sie erstellt haben.



3. Öffnen Sie den Stack und zeigen Sie den Layer und die Instance an. Beachten Sie, dass die Ebene und die Instanz mit den Namen und anderen Metadaten erstellt wurden, die in der AWS CloudFormation Vorlage angegeben sind. Sie können nun Ihren Stack und Layer für die Verwendung benutzerdefinierter Chef-Rezeptbücher und -Rezepte konfigurieren.

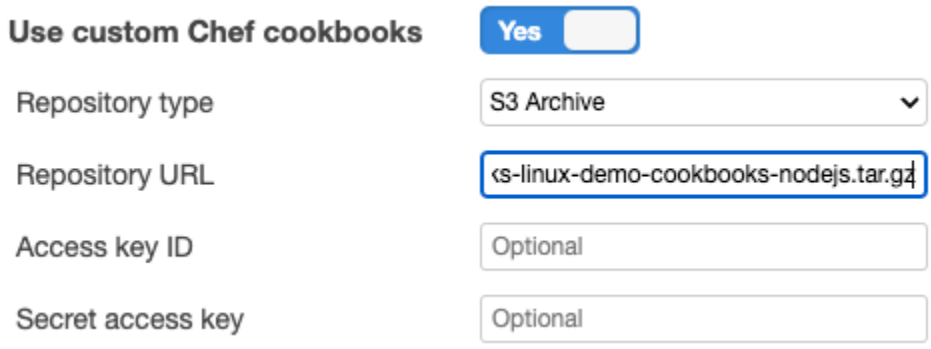
Schritt 2: Konfigurieren von Stack und Layer für die Verwendung von benutzerdefinierten Rezeptbüchern

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Chef 12 Stacks in AWS OpsWorks Stacks benötigen Ihre eigenen oder von der Community erstellten Kochbücher, um benutzerdefinierte Anwendungsebenen zu erstellen. Für die Zwecke dieser Anleitung können Sie auf ein Repository verweisen, das bereits eine Reihe von Chef-Rezeptbüchern und Chef-Rezepten enthält. Über diese Rezepte werden das Node.js-Paket sowie dessen Abhängigkeiten auf Ihrer Instance installiert. Mit anderen Chef-Rezepten können Sie die Node.js-Anwendung bereitstellen, die Sie in Schritt 4: Füge deine App zu AWS OpsWorks Stacks hinzu vorbereiten werden. Das Chef-Rezept, das Sie in diesem Schritt angeben, wird jedes Mal ausgeführt, wenn eine neue Version Ihrer Anwendung von bereitgestellt wird. CodePipeline

1. Öffnen Sie in der AWS OpsWorks Stacks-Konsole den Stack, in Schritt 1: Erstellen Sie einen Stapel, eine Ebene und eine Instanz in AWS OpsWorks Stacks dem Sie ihn erstellt haben. Wählen Sie Stack Settings (Stack-Einstellungen) und anschließend Edit (Bearbeiten) aus.
2. Legen Sie Use custom Chef Cookbooks (Benutzerdefinierte Chef-Rezeptbücher verwenden) auf Yes (Ja) fest, um die zugehörigen benutzerdefinierten Rezeptbucheinstellungen anzuzeigen.
3. Wählen Sie aus der Dropdown-Liste Repository type (Repository-Typ) die Option S3 Archive (S3-Archiv) aus. Um sowohl mit als auch CodePipeline arbeiten zu können AWS OpsWorks, muss Ihre Kochbuchquelle S3 sein.
4. Geben Sie für Repository URL (Repository-URL) **`https://s3.amazonaws.com/opsworks-demo-assets/opsworks-linux-demo-cookbooks-nodejs.tar.gz`** an. Die Einstellungen sollten in etwa wie folgt aussehen:



5. Wählen Sie Speichern.
6. Wählen Sie im Navigationsbereich Ebenen aus.
7. Wählen Sie Settings (Einstellungen) für den Layer aus, den Sie in Schritt 1: Erstellen Sie einen Stapel, eine Ebene und eine Instanz in AWS OpsWorks Stacks erstellt haben.

8. Stellen Sie auf der Registerkarte General Settings (Allgemeine Einstellungen) sicher, dass der Name des Layers Node.js App Server und der Kurzname des Layers app1 ist. Wählen Sie Recipes (Rezepte) aus.
9. Legen Sie auf der Registerkarte Recipes (Rezepte) **`nodejs_demo`** als Rezept fest, das während des Lebenszyklusereignisses Deploy (Bereitstellen) ausgeführt werden soll. Wählen Sie Speichern.
10. Wählen Sie auf der Registerkarte Sicherheit aus der Drop-down-Liste Sicherheitsgruppen die Sicherheitsgruppe AWS- OpsWorks -Webapp aus.
11. Wählen Sie Speichern.

Schritt 3: App-Code in einen Amazon S3 S3-Bucket hochladen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

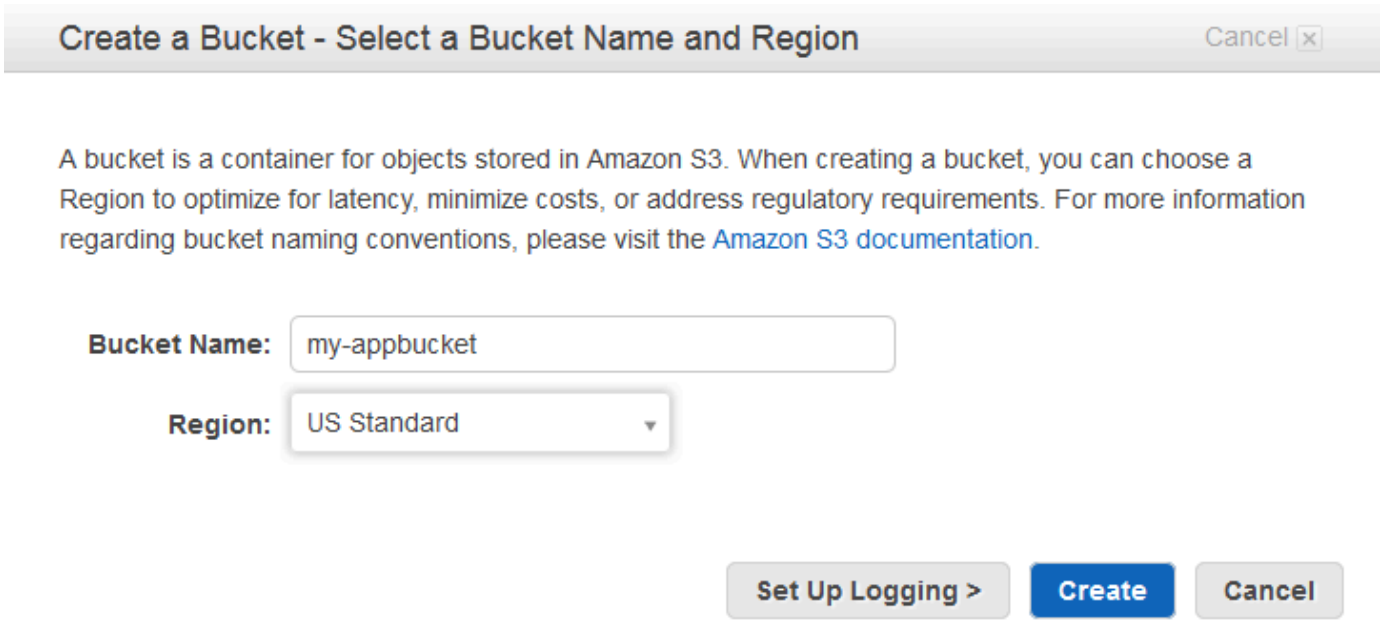
Da Sie einen Link zu Ihrem Code-Repository als Teil der Pipeline-Einrichtung angeben müssen, halten Sie das Code-Repository bereit, bevor Sie Ihre Pipeline erstellen. In dieser exemplarischen Vorgehensweise laden Sie eine App Node.js in einen Amazon S3 S3-Bucket hoch.

Obwohl Code direkt aus GitHub oder CodeCommit als Quellen verwendet werden CodePipeline kann, zeigt diese exemplarische Vorgehensweise, wie Sie einen Amazon S3 S3-Bucket verwenden. In dieser exemplarischen Vorgehensweise laden Sie die Beispiel-App Node.js in Ihren eigenen Amazon S3 S3-Bucket hoch, damit Sie Änderungen an der App vornehmen können. Der Amazon S3 S3-Bucket, den Sie in diesem Schritt erstellen CodePipeline , ermöglicht es, Änderungen am App-Code zu erkennen und die geänderte App automatisch bereitzustellen. Sie können auch einen vorhandenen Bucket verwenden. Stellen Sie sicher, dass der Bucket die in Simple Pipeline Walkthrough (Amazon S3 Bucket) in der CodePipeline Dokumentation beschriebenen Kriterien erfüllt.

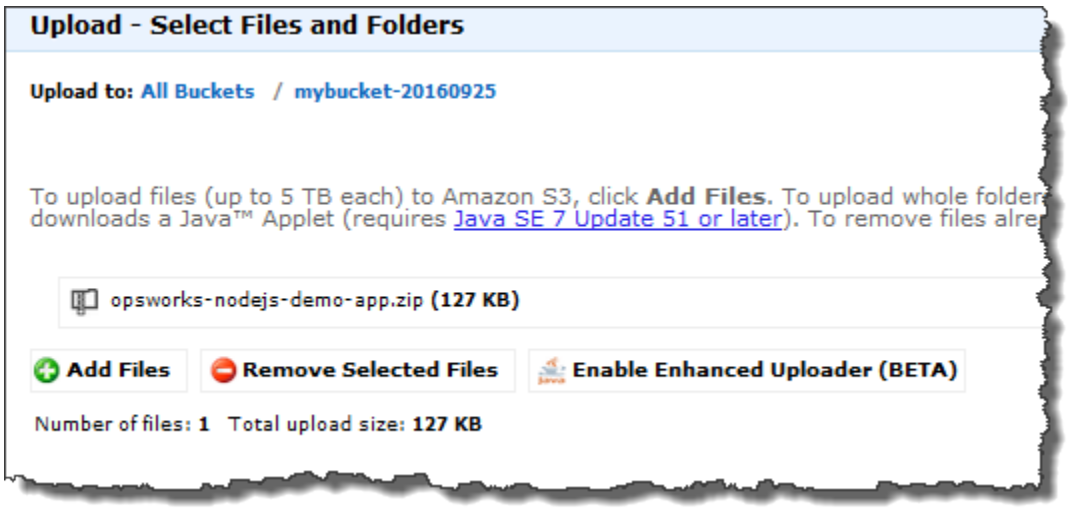
> **Important**
>
> Der Amazon S3 S3-Bucket muss sich in derselben Region befinden, in der Sie später Ihre Pipeline erstellen werden. CodePipeline Unterstützt derzeit nur den AWS OpsWorks Stacks-Anbieter in der Region USA Ost (Nord-Virginia) (us-east-1). Alle Ressourcen in dieser exemplarischen Vorgehensweise sollten in der Region USA Ost (Nord-Virginia) erstellt werden. Der Bucket muss auch versioniert sein, da eine versionierte CodePipeline Quelle erforderlich ist. Weitere Informationen finden Sie unter Verwenden der Versionsverwaltung.

So laden Sie Ihre App in einen Amazon S3 S3-Bucket hoch

1. Laden Sie die ZIP-Datei der AWS OpsWorks Stacks-Beispiel-App, Node.js, herunter und speichern Sie sie an einem geeigneten Ort auf Ihrem lokalen Computer.
2. Öffnen Sie die Amazon-S3-Konsole unter https://console.aws.amazon.com/s3/.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.
4. Geben Sie auf der Seite Create a Bucket - Select a Bucket Name and Region (Bucket erstellen – Bucket-Namen und Region auswählen) für Bucket Name (Bucket-Name) einen eindeutigen Namen für den Bucket ein. Bucket-Namen müssen für alle AWS Konten eindeutig sein, nicht nur für Ihr eigenes Konto. In dieser Anleitung verwenden wir den Namen **`my-appbucket`**. Sie können als eindeutigen Bucket-Namen aber auch `my-appbucket-`*`yearmonthday`* verwenden. Wählen Sie aus der Dropdown-Liste Region die Option US Standard und anschließend Create (Erstellen) aus. US Standard entspricht `us-east-1`.



5. Wählen Sie in der Liste All Buckets (Alle Buckets) den von Ihnen erstellten Bucket aus.
6. Wählen Sie auf der Bucket-Seite Upload (Hochladen) aus.
7. Wählen Sie auf der Seite Upload - Select Files and Folders (Hochladen – Dateien und Ordner auswählen) die Option Add Files (Dateien hinzufügen) aus. Suchen Sie nach der ZIP-Datei, die Sie in Schritt 1 gespeichert haben, wählen Sie Open (Öffnen) und anschließend Start Upload (Hochladen starten) aus.



8. Nachdem Sie die Datei hochgeladen haben, wählen Sie die ZIP-Datei aus der Dateiliste in Ihrem Bucket aus und klicken Sie dann auf Properties (Eigenschaften).

9. Kopieren Sie im Bereich Properties (Eigenschaften) den Link zu Ihrer ZIP-Datei und notieren Sie den Link. Sie benötigen den Bucket-Namen und den Teil des Namens der ZIP-Datei dieses Links, um Ihre Pipeline zu erstellen.

Schritt 4: Füge deine App zu AWS OpsWorks Stacks hinzu

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

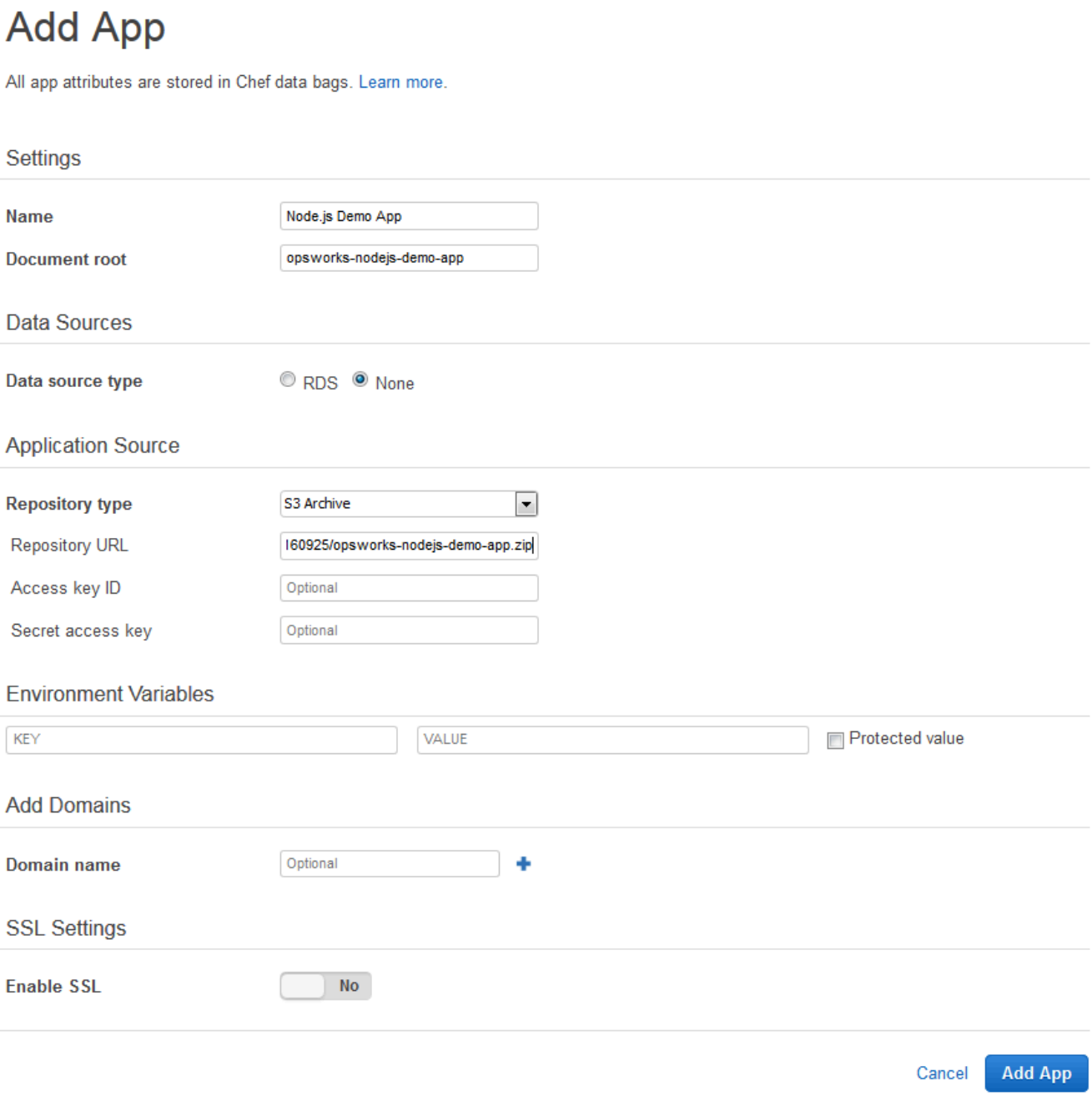
Bevor Sie eine Pipeline in erstellen CodePipeline, fügen Sie die Test-App Node.js zu AWS OpsWorks Stacks hinzu. Wenn Sie die Pipeline erstellen, müssen Sie die App auswählen, die Sie zu AWS OpsWorks Stacks hinzugefügt haben.

Halten Sie den Amazon S3 S3-Bucket-Link aus Schritt 9 des vorherigen Verfahrens bereit. Sie benötigen den Link zum Bucket, auf den Sie Ihre Testanwendung gespeichert haben, um diese Anleitung abzuschließen.

Um eine App zu AWS OpsWorks Stacks hinzuzufügen

1. Öffnen CodePipelineDemoSie in der AWS OpsWorks Stacks-Konsole und wählen Sie im Navigationsbereich Apps aus.
2. Wählen Sie Add app (App hinzufügen) aus.
3. Geben Sie auf der Seite Add App (App hinzufügen) die folgende Information an:
   a. Geben Sie einen Namen für Ihre Anwendung an. In dieser Anleitung wird der Name `Node.js Demo App` verwendet.
   b. Wählen Sie für Data source type (Datenquellentyp) die Option None (Kein) aus. Diese Anwendung erfordert keine externe Datenbank oder Datenquelle.
   c. Wählen Sie aus der Dropdown-Liste Repository type (Repository-Typ) die Option S3 Archive (S3-Archiv) aus.

d. Fügen Sie in das Textfeld Repository URL (Repository-URL) die URL ein, die Sie in Schritt 9 von Schritt 3: App-Code in einen Amazon S3 S3-Bucket hochladen kopiert haben. Ihr Formular sollte ähnlich wie folgt aussehen:



4. Sie müssen in diesem Formular keine weiteren Einstellungen ändern. Wählen Sie Add App (Anwendung hinzufügen) aus.
5. Wenn die App Node.js Demo App (Node.js Demo-App) in der Liste auf der Seite Apps angezeigt wird, fahren Sie mit der nächsten Anleitung fort, Schritt 5: Erstellen Sie eine Pipeline in CodePipeline.

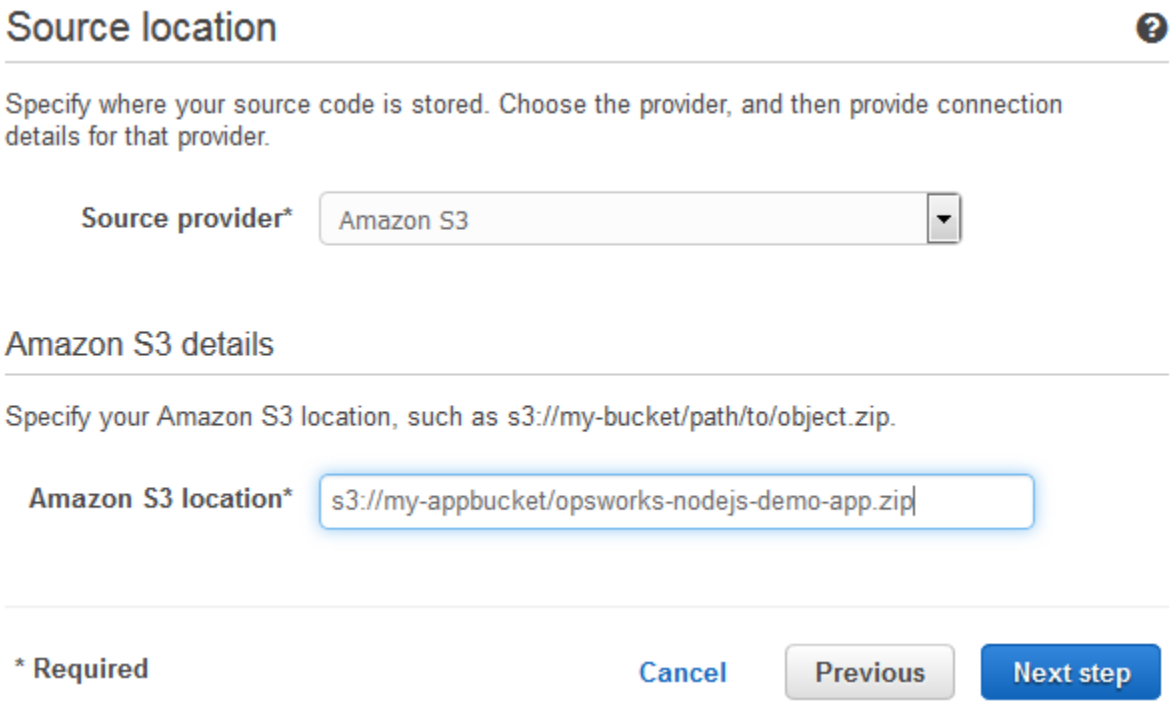
Schritt 5: Erstellen Sie eine Pipeline in CodePipeline

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

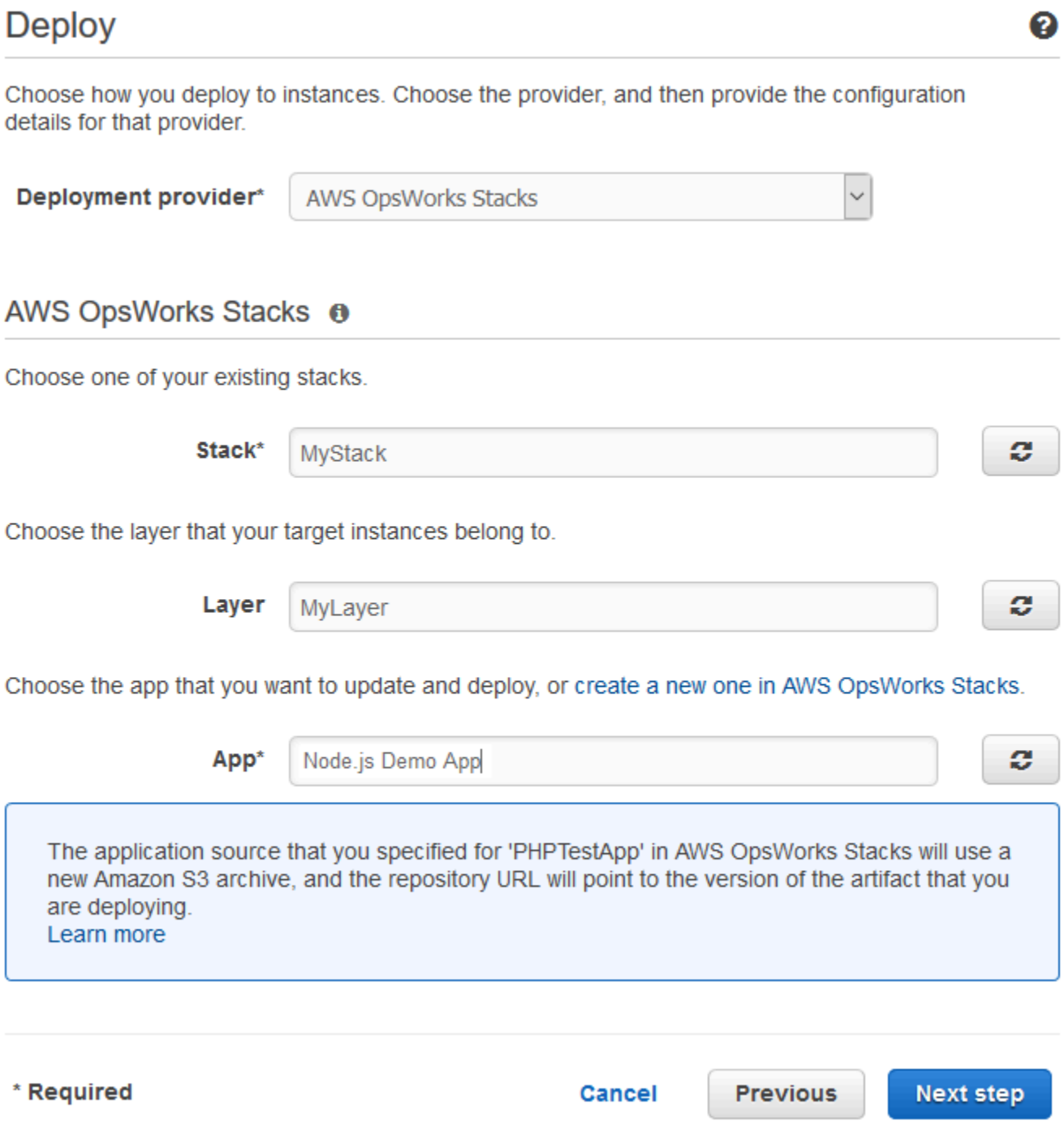
Nachdem Sie einen Stack mit einer Ebene und mindestens einer Instanz in AWS OpsWorks Stacks konfiguriert haben, erstellen Sie eine Pipeline CodePipeline mit AWS OpsWorks Stacks als Anbieter, um Apps oder Chef-Kochbücher für Ihre Stacks-Ressourcen bereitzustellen. AWS OpsWorks

So erstellen Sie eine Pipeline

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/codepipeline/ CodePipeline .
2. Wählen Sie Create pipeline (Pipeline erstellen) aus.
3. Geben **`MyOpsWorksPipeline`** Sie auf der CodePipeline Seite Erste Schritte mit einen beliebigen anderen Pipelinenamen ein, der für Ihr Konto eindeutig ist, und wählen Sie dann Weiter aus.
4. Wählen Sie auf der Seite Source Location (Quellspeicherort) die Option Amazon S3 aus der Dropdown-Liste Source provider (Quellanbieter) aus.
5. Geben Sie im Bereich Amazon S3 S3-Details Ihren Amazon S3 S3-Bucket-Pfad im folgenden Format ein**`s3://`*`bucket-name`*`/`*`file name`***. Verwenden Sie dabei den Link, den Sie in Schritt 9 von Schritt 3: App-Code in einen Amazon S3 S3-Bucket hochladen notiert haben. In dieser Anleitung ist der Pfad `s3://my-appbucket/opsworks-nodejs-demo-app.zip`. Klicken Sie auf Nächster Schritt.



6. Wählen Sie auf der Seite Build die Option No Build (Kein Build) aus der Dropdown-Liste und anschließend Next step (Nächster Schritt) aus.
7. Wählen Sie auf der Seite Deploy (Bereitstellen) als Bereitstellungsanbieter AWS OpsWorks Stacks aus.



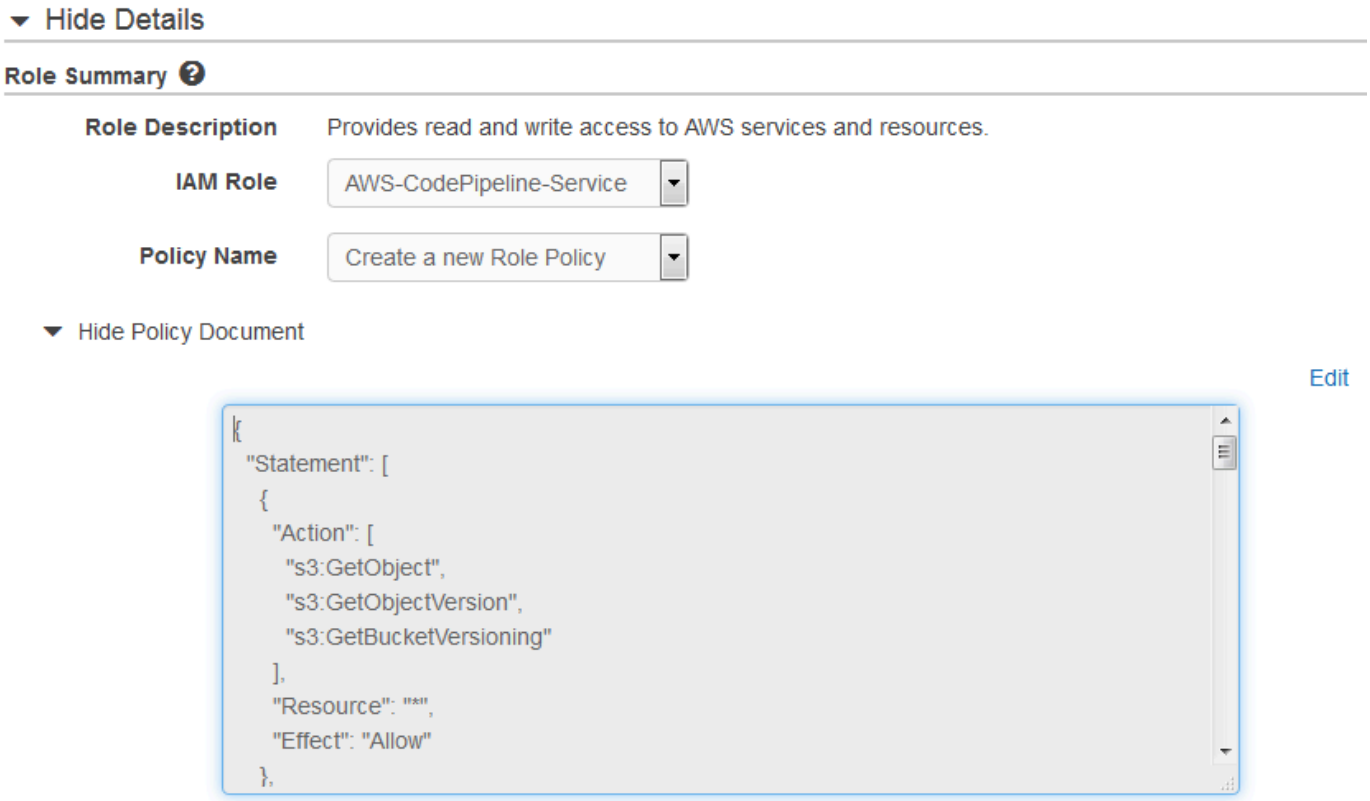
8. Geben Sie im Feld Stack `CodePipelineDemo` oder den Namen des Stacks ein, den Sie in Schritt 1: Erstellen Sie einen Stapel, eine Ebene und eine Instanz in AWS OpsWorks Stacks erstellt haben.
9. Geben Sie im Feld Layer `Node.js App Server` oder den Namen des Layers ein, den Sie in Schritt 1: Erstellen Sie einen Stapel, eine Ebene und eine Instanz in AWS OpsWorks Stacks erstellt haben.

10. Wählen Sie im Feld App die App aus, in der Sie auf Amazon S3 hochgeladen habenSchritt 3: App-Code in einen Amazon S3 S3-Bucket hochladen, und wählen Sie dann Nächster Schritt aus.
11. Wählen Sie auf der Seite AWS Service Role die Option Create Role aus.

    Es öffnet sich ein neues Fenster mit einer IAM-Konsolenseite, auf der die Rolle beschrieben wird, die für Sie erstellt wird. `AWS-CodePipeline-Service` Wählen Sie aus der Dropdown-Liste Policy name (Richtlinienname) die Option Create new policy (Neue Richtlinie erstellen) aus. Stellen Sie sicher, dass das Richtliniendokument den folgenden Inhalt hat. Wählen Sie Edit (Bearbeiten) aus, um gegebenenfalls das Richtliniendokument zu ändern.

    ```
    {
        "Statement": [
            {
                "Action": [
                    "s3:GetObject",
                    "s3:GetObjectVersion",
                    "s3:GetBucketVersioning"
                ],
                "Resource": "*",
                "Effect": "Allow"
            },
            {
                "Action": "opsworks:*",
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    }
    ```

    Wenn Sie alle gewünschten Änderungen für das Richtliniendokument ausgeführt haben, wählen Sie Allow (Zulassen) aus. Ihre Änderungen werden in der IAM-Konsole angezeigt.



> **Note**
>
> Wenn die Rollenerstellung fehlschlägt, liegt das möglicherweise daran, dass Sie bereits über eine IAM-Rolle mit dem Namen CodePipelineAWS-Service verfügen. Wenn Sie die Rolle AWS- CodePipeline -Service vor Mai 2016 verwendet haben, verfügt die Rolle möglicherweise nicht über die Berechtigungen, AWS OpsWorks Stacks als Bereitstellungsanbieter zu verwenden. Sie müssen in diesem Fall die Richtlinienanweisung wie in diesem Schritt beschrieben aktualisieren. Wenn Ihnen eine Fehlermeldung angezeigt wird, kehren Sie zum Anfang dieses Schritts zurück und wählen anstelle von Create role (Rolle erstellen) die Option Use existing role (Vorhandene Rolle verwenden) aus. Wenn Sie eine vorhandene Rolle verwenden, sollte die Rolle einer Richtlinie zugewiesen sein, die die Berechtigungen, wie in diesem Schritt dargestellt, enthält. Weitere Informationen zur Servicerolle und deren Richtlinienanweisung finden Sie unter Bearbeiten einer Richtlinie für eine IAM-Servicerolle.

12. Wenn der Prozess zur Rollenerstellung erfolgreich ist, wird die IAM-Seite geschlossen und Sie kehren zur Seite mit der AWS Servicerolle zurück. Klicken Sie auf Nächster Schritt.

13. Überprüfen Sie Ihre Auswahl auf der Seite Review your pipeline (Ihre Pipeline überprüfen) und wählen Sie dann Create pipeline (Pipeline erstellen) aus.
14. Wenn Ihre Pipeline bereit ist, sollte sie automatisch damit beginnen. Ihren Quellcode zu ermitteln und Ihre Anwendung zu Ihrem Stack bereitzustellen. Dieser Vorgang kann einige Minuten dauern.

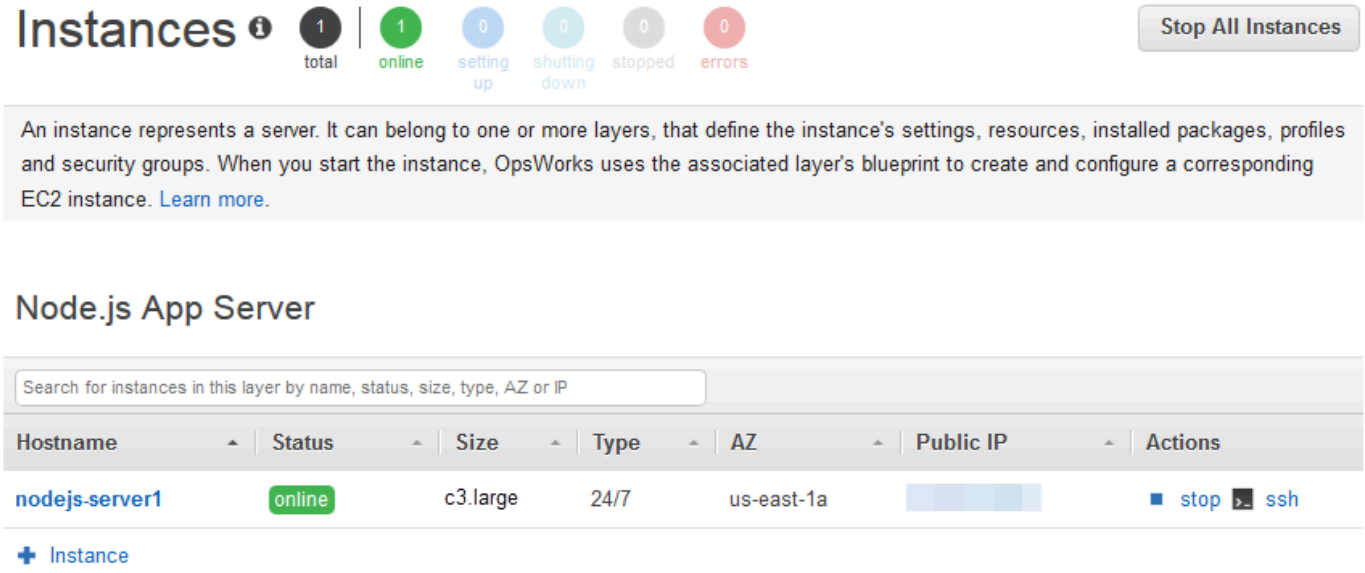
Schritt 6: Überprüfen der App-Bereitstellung in AWS OpsWorks Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

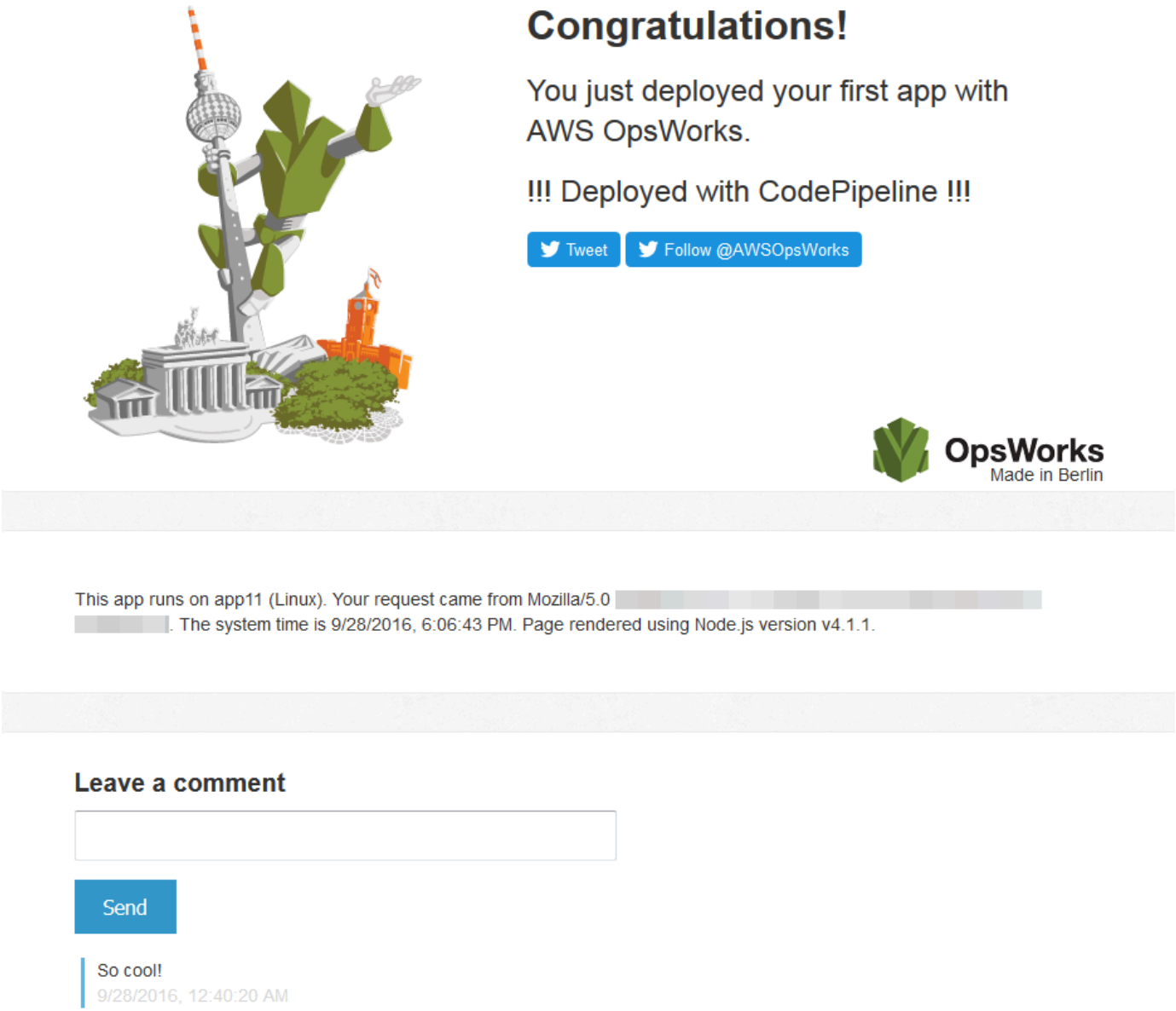
Um zu überprüfen, ob die App Node.js in Ihrem Stack CodePipeline bereitgestellt wurde, melden Sie sich bei der Instanz an, in der Sie sie erstellt haben. Schritt 1: Erstellen Sie einen Stapel, eine Ebene und eine Instanz in AWS OpsWorks Stacks Hier sollten Sie die Node.js-Webanwendung sehen und verwenden können.

Um die App-Bereitstellung in Ihrer AWS OpsWorks Stacks-Instanz zu überprüfen

1. Öffnen Sie die AWS OpsWorks Konsole unter https://console.aws.amazon.com/opsworks/.
2. Wählen Sie im AWS OpsWorks Stacks-Dashboard CodePipelineDemodie Option Node.js App Server aus.
3. Wählen Sie im Navigationsbereich Instances und anschließend die öffentliche IP-Adresse der Instance aus, die Sie erstellt haben, um die Webanwendung anzuzeigen.



Die Anwendung wird auf einer neuen Registerkarte angezeigt werden.



Schritt 7 (optional): Aktualisieren Sie den App-Code, um zu sehen, wie Ihre App CodePipeline automatisch erneut bereitgestellt wird

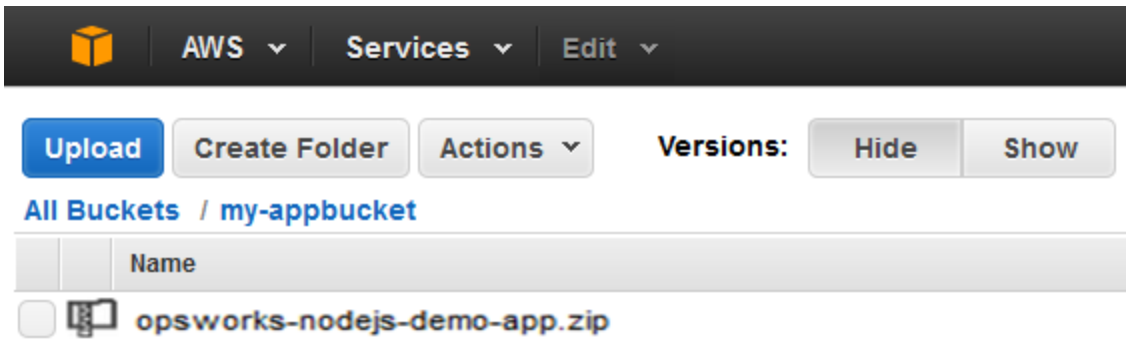
> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn Sie Änderungen am Code in Apps oder Cookbooks vornehmen, die Sie mithilfe von Using bereitgestellt haben CodePipeline, werden die aktualisierten Artefakte automatisch CodePipeline auf Ihren Zielinstanzen (in diesem Fall auf einem AWS OpsWorks Ziel-Stacks-Stack) bereitgestellt. In diesem Abschnitt wird beschrieben, wie die App automatisch erneut bereitgestellt wird, wenn Sie den Code in Ihrer Beispiel-App Node.js aktualisieren. Wenn Sie den App-Code für diese Anleitung noch lokal gespeichert haben und seit dem Beginn der Anleitung niemand Änderungen am Code vorgenommen hat, können Sie die Schritte 1 bis 4 dieser Anleitung überspringen.

So bearbeiten Sie den Code in der Beispielanwendung

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/s3/.
2. Öffnen Sie den Bucket, in dem Sie die Beispiel-App Node.js speichern.

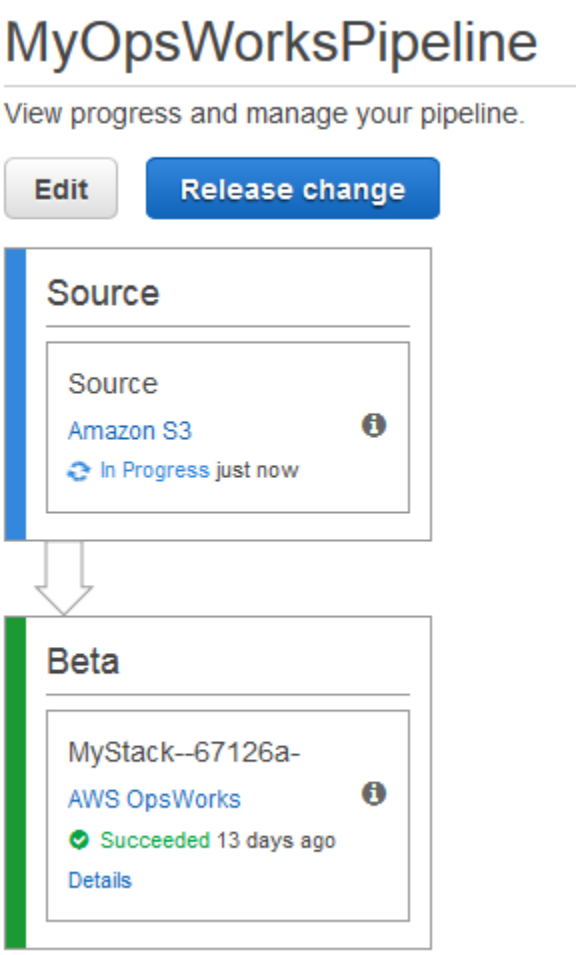


3. Wählen Sie die ZIP-Datei, die die Anwendung enthält. Wählen Sie im Menü Actions die Option Download aus.
4. Öffnen Sie im Dialogfeld mit der rechten Maustaste das Kontextmenü, wählen Sie Download (Herunterladen) aus und speichern Sie dann die ZIP-Datei an einem geeigneten Ort. Wählen Sie OK aus.
5. Extrahieren Sie die Inhalte der ZIP-Datei an einem geeigneten Ort. Möglicherweise müssen Sie Berechtigungen für die extrahierten Ordner und deren Unterordner und Inhalte ändern, sodass eine Bearbeitung zugelassen wird. Öffnen Sie im Ordner `opsworks-nodejs-demo-app\views` die Datei `header.html`, um sie zu bearbeiten.
6. Suchen Sie nach der Zeichenfolge `You just deployed your first app with`. Ersetzen Sie das Wort `deployed` durch `updated`. Ändern Sie in der nächsten Zeile `AWS OpsWorks.` in `AWS OpsWorks and AWS CodePipeline.`. Bearbeiten Sie nur den Text.

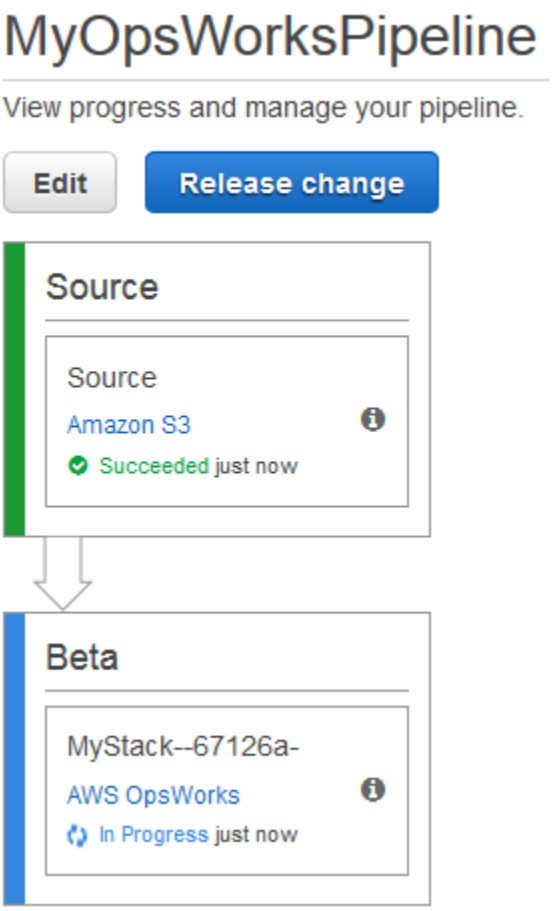
```
<div id="main" role="main">LF
  <div class="container">LF
    <div class="hero-unit">LF
      <div class="robot">LF
        <h1>Congratulations!</h1>LF
        <h2>LF
          You just updated your first app with<br/>LF
          AWS OpsWorks and AWS CodePipeline.LF
        </h2>LF
```

7. Speichern und schließen Sie die Datei `header.html`.
8. Packen Sie das Verzeichnis `opsworks-nodejs-demo-app` und speichern Sie die ZIP-Datei. Ändern Sie nicht den Namen der ZIP-Datei.
9. Laden Sie die neue ZIP-Datei in Ihren Amazon S3 S3-Bucket hoch. In dieser Anleitung ist der Name des Buckets `my-appbucket`.
10. Öffnen Sie die CodePipeline Konsole und öffnen Sie Ihre AWS OpsWorks Stacks-Pipeline (MyOpsWorksPipeline). Wählen Sie Release Change (Versionsänderung) aus.

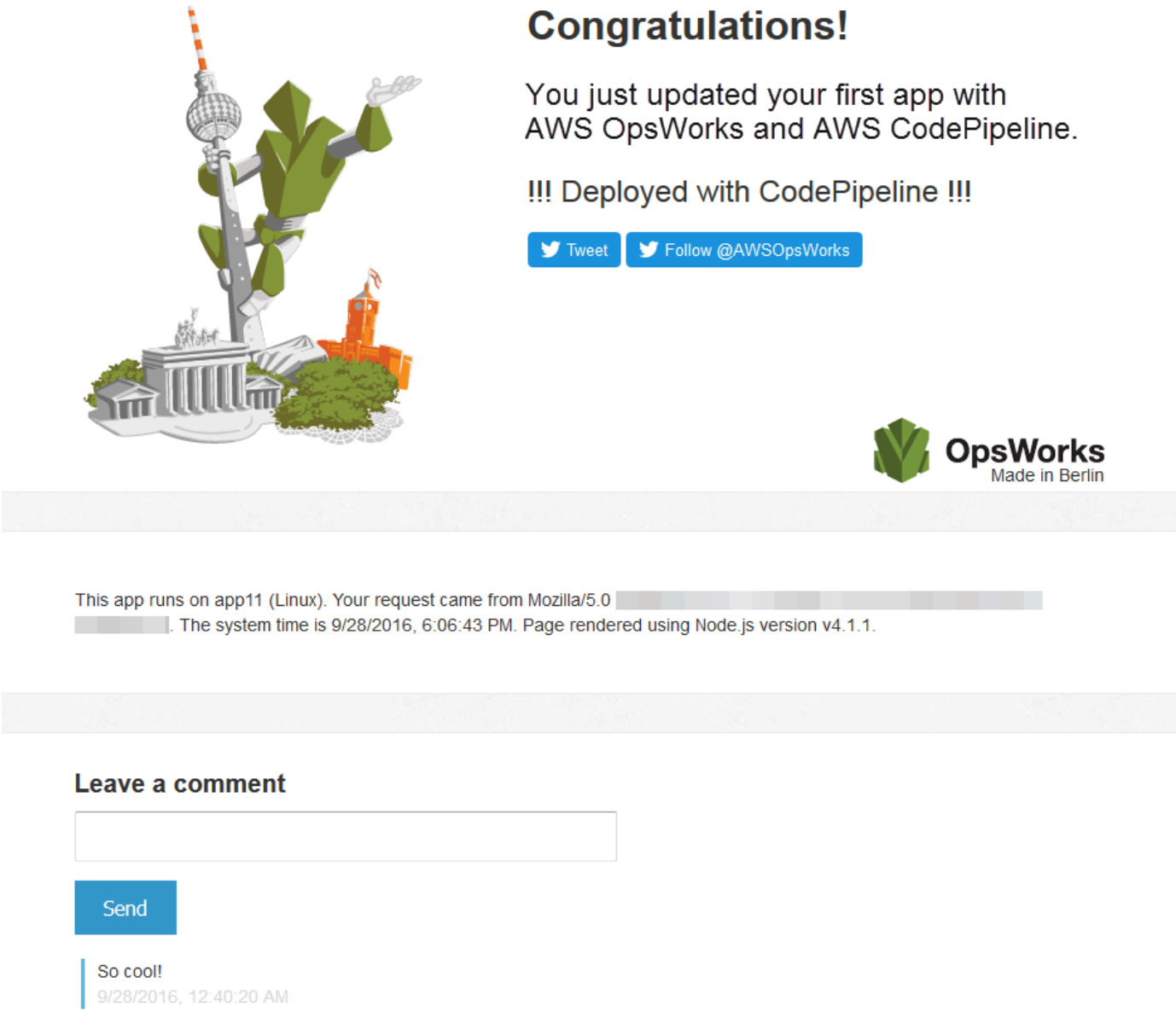
    (Sie können warten CodePipeline , bis Sie die Codeänderung aus der aktualisierten Version der App in Ihrem Amazon S3 S3-Bucket feststellen. Um Ihnen Zeit zu sparen, werden Sie in dieser exemplarischen Vorgehensweise aufgefordert, einfach Release Change auszuwählen.)
11. Beobachten Sie, CodePipeline wie die einzelnen Phasen der Pipeline durchlaufen werden. CodePipeline Erkennt zunächst Änderungen am Quellartefakt.



CodePipeline verschiebt den aktualisierten Code auf Ihren Stack in AWS OpsWorks Stacks.



12. Wenn beide Phasen der Pipeline erfolgreich abgeschlossen wurden, öffnen Sie Ihren Stack in AWS OpsWorks Stacks.
13. Wählen Sie auf der Eigenschaftsseite des Stacks Instances aus.
14. Wählen Sie in der Spalte Public IP (Öffentliche IP-Adresse) die öffentliche IP-Adresse Ihrer Instance aus, um den Text der aktualisierten Anwendung anzuzeigen.



Schritt 8 (optional): Bereinigen von Ressourcen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Um zu verhindern, dass Ihr AWS Konto ungewollt belastet wird, können Sie die AWS Ressourcen löschen, die Sie für diese exemplarische Vorgehensweise verwendet haben. Zu diesen AWS Ressourcen gehören der AWS OpsWorks Stacks-Stack, die IAM-Rolle und das Instanzprofil sowie die Pipeline, in der Sie erstellt haben. CodePipeline Möglicherweise möchten Sie diese AWS Ressourcen jedoch weiterhin verwenden, wenn Sie mehr über AWS OpsWorks Stacks und erfahren. CodePipeline Wenn Sie diese Ressourcen behalten möchten, haben Sie diese Anleitung abgeschlossen.

So löschen Sie die Anwendung aus dem Stack

Da Sie die App nicht als Teil Ihrer AWS CloudFormation Vorlage erstellt oder angewendet haben, löschen Sie die Test-App Node.js, bevor Sie den Stack in AWS CloudFormation löschen.

1. Wählen Sie in der AWS OpsWorks Stacks-Konsole im Navigationsbereich des Dienstes Apps aus.
2. Wählen Sie auf der Seite Apps die Option Node.js Demo App (Node.js-Demo-App) und anschließend in Actions (Aktionen) die Option delete (löschen) aus. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Löschen. AWS OpsWorks Stacks löscht die App.

So löschen Sie den Stack

Da Sie den Stack erstellt haben, indem Sie eine AWS CloudFormation Vorlage ausgeführt haben, können Sie den Stack, einschließlich der Ebene, der Instanz, des Instanzprofils und der Sicherheitsgruppe, die die Vorlage erstellt hat, in der AWS CloudFormation Konsole löschen.

1. Öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie im AWS CloudFormation Konsolen-Dashboard den Stack aus, den Sie erstellt haben. Wählen Sie im Menü Actions (Aktionen) die Option Delete Stack (Stack löschen) aus. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Yes, Delete (Ja, löschen) aus.
3. Warten Sie, bis DELETE_COMPLETE in der Spalte Status (Status) für den Stack angezeigt wird.

So löschen Sie die Pipeline

1. Öffnen Sie die CodePipeline Konsole.

2. Wählen Sie im CodePipeline Dashboard die Pipeline aus, die Sie für diese exemplarische Vorgehensweise erstellt haben.
3. Wählen Sie auf der Pipeline-Seite Edit (Bearbeiten) aus.
4. Klicken Sie auf der Seite Edit auf Delete. Wenn Sie aufgefordert werden, Ihre Entscheidung zu bestätigen, wählen Sie Delete aus.

AWS CodePipeline mit AWS OpsWorks Stacks - Chef 11 Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS CodePipelineermöglicht es Ihnen, Continuous-Delivery-Pipelines zu erstellen CodeCommit, die Codeänderungen aus Quellen wie Amazon Simple Storage Service (Amazon S3) oder GitHubverfolgen. Das Beispiel in diesem Thema beschreibt, wie Sie eine einfache Pipeline CodePipeline als Bereitstellungstool für Code erstellen und verwenden, den Sie auf AWS OpsWorks Stacks-Layern ausführen. In diesem Beispiel erstellen Sie eine Pipeline für eine einfache PHP-App und weisen AWS OpsWorks Stacks dann an, die App auf allen Instanzen in einer Ebene in einem Chef 11.10-Stack (in diesem Fall einer einzelnen Instanz) auszuführen.

> **Note**
>
> Dieses Thema beschreibt, wie Sie eine Pipeline nutzen, um eine Anwendung auf Chef 11.10-Stack auszuführen und zu aktualisieren. Für Informationen wie Sie eine Pipeline nutzen, um eine Anwendung auf Chef 11.10-Stack auszuführen und zu aktualisieren, lesen Sie AWS CodePipeline mit AWS OpsWorks Stacks - Chef 12 Stacks. Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Themen

Voraussetzungen

Bevor Sie diese Anleitung starten, stellen Sie sicher, dass Sie über Administratorberechtigungen verfügen, um alle folgenden Aufgaben auszuführen. Sie können Mitglied einer Gruppe sein, auf die die AdministratorAccessRichtlinie angewendet wurde, oder Sie können Mitglied einer Gruppe sein, die über die in der folgenden Tabelle aufgeführten Berechtigungen und Richtlinien verfügt. Aus Sicherheitsgründen sollten Sie einer Gruppe angehören, die über die erforderlichen Rechte für die folgenden Aufgaben verfügt, anstatt einzelnen Benutzern die erforderlichen Berechtigungen zuzuweisen.

Weitere Informationen zum Erstellen einer Sicherheitsgruppe in IAM und zum Zuweisen von Berechtigungen zu dieser Gruppe finden Sie unter IAM-Benutzergruppen erstellen. Weitere Informationen zur Verwaltung von AWS OpsWorks Stacks-Berechtigungen finden Sie unter Bewährte Methoden: Berechtigungen verwalten.

Berechtigungen	Empfohlene Richtlinie für das Anfügen an eine Gruppe
Erstellen und bearbeiten Sie Stapel, Ebenen und Instanzen in AWS OpsWorks Stacks.	AWSOpsWorks_FullAccess
Erstellen, bearbeiten und führen Sie AWS CloudFormation-Vorlagen aus.	AmazonCloudFormationFullAccess

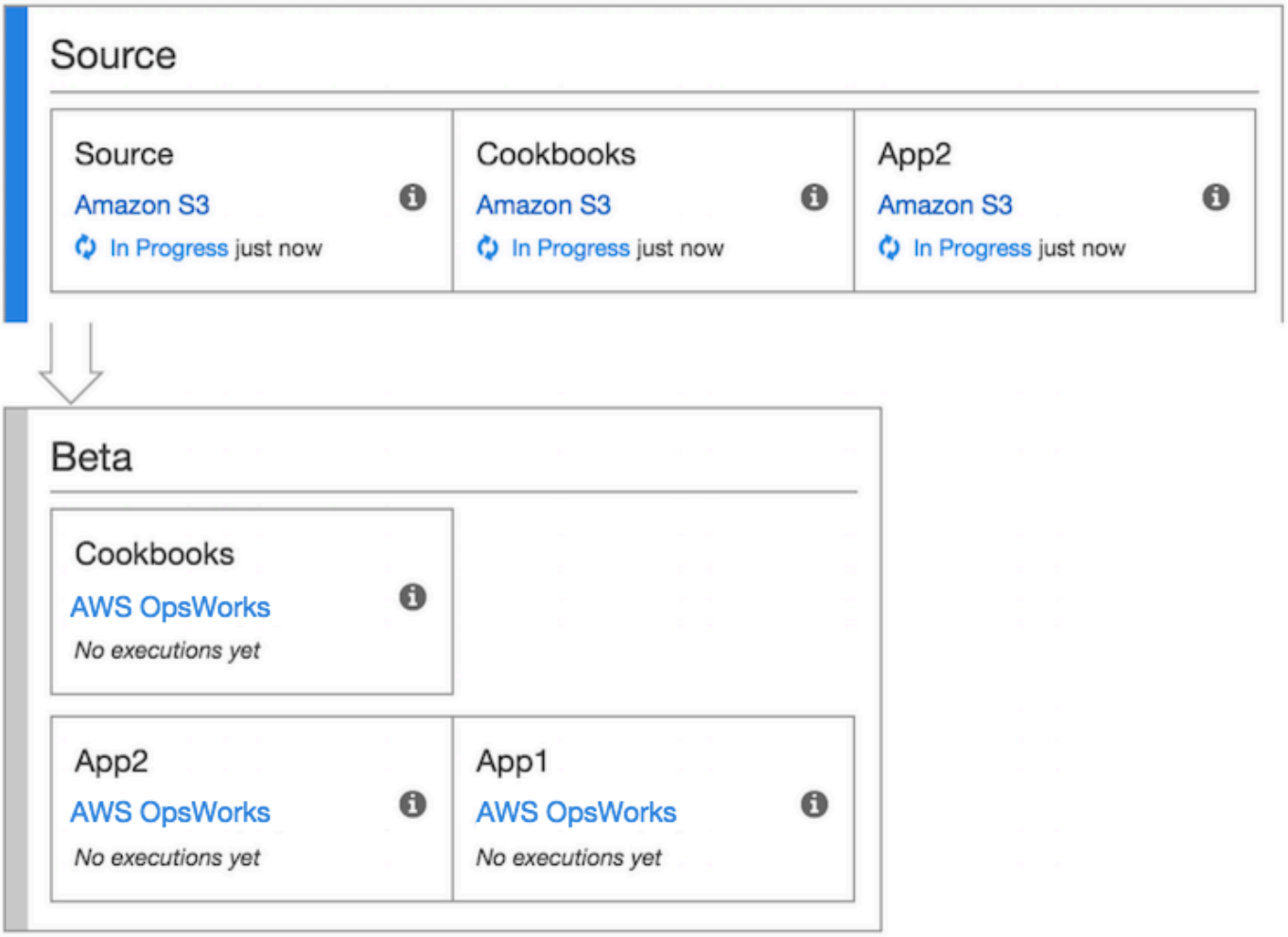
Berechtigungen	Empfohlene Richtlinie für das Anfügen an eine Gruppe
Erstellen, bearbeiten und greifen Sie auf Amazon S3 S3-Buckets zu.	Amazon S3 FullAccess
Pipelines erstellen, bearbeiten und ausführen , insbesondere in Pipelines CodePipeline, die AWS OpsWorks Stacks als Anbieter verwenden.	AWSCodePipeline_FullAccess

Sie benötigen außerdem ein Amazon EC2 EC2-Schlüsselpaar. Sie werden aufgefordert, den Namen dieses key pair anzugeben, wenn Sie die AWS CloudFormation Vorlage ausführen, mit der der Beispielstapel, die Ebene und die Instanz in dieser exemplarischen Vorgehensweise erstellt werden. Weitere Informationen zum Abrufen eines key pair in der Amazon EC2 EC2-Konsole finden Sie unter Create a Key Pair in der Amazon EC2 EC2-Dokumentation. Das key pair sollte sich in der Region USA Ost (Nord-Virginia) befinden. Sie können ein vorhandenes Schlüsselpaar verwenden, wenn Sie in der betreffenden Region bereits über ein Schlüsselpaar verfügen.

Andere unterstützte Szenarien

Diese Anleitung erstellt eine einfache Pipeline, die die Stufen Source (Quelle) und Deploy (Bereitstellen) umfasst. Sie können jedoch komplexere Pipelines erstellen, die AWS OpsWorks Stacks als Anbieter verwenden. Im Folgenden werden einige Beispiele für unterstützte Pipelines und Szenarien aufgeführt:

- Sie können eine Pipeline bearbeiten, um ein Chef-Rezeptbuch der Stufe Source (Quelle) und ein zugehöriges Ziel für aktualisierte Rezeptbücher der Stufe Deploy (Bereitstellen) hinzuzufügen. In diesem Fall fügen Sie eine Deploy (Bereitstellen)-Aktion hinzu, die eine Aktualisierung Ihrer Rezeptbücher auslöst, wenn Sie Änderungen an der Quelle vornehmen. Das aktualisierte Rezeptbuch wird vor Ihrer Anwendung bereitgestellt.
- Sie können eine komplexe Pipeline mit benutzerdefinierten Kochbüchern und mehreren Apps erstellen und diese in einem AWS OpsWorks Stacks-Stack bereitstellen. Die Pipeline verfolgt Änderungen an der Anwendung und den Rezeptbuchquellen und stellt sich erneut bereit, wenn Sie Änderungen vorgenommen haben. Die folgende Abbildung zeigt ein Beispiel einer ähnlichen, komplexen Pipeline:



Weitere Informationen zur Arbeit mit CodePipeline finden Sie in der CodePipeline Dokumentation.

Schritt 1: Erstellen eines Stacks, Layers und einer Instance in AWS OpsWorks Stacks

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Um AWS OpsWorks Stacks als Bereitstellungsanbieter für eine Pipeline verwenden zu können, müssen Sie zunächst über einen Stack, eine Ebene und mindestens eine Instanz in der Ebene verfügen. Sie können zwar einen Stack in AWS OpsWorks Stacks erstellen, indem Sie den Anweisungen unter Erste Schritte mit Linux Stacks oder Erste Schritte mit Windows Stacks folgen. Um Zeit zu sparen, verwendet dieses Beispiel jedoch eine AWS CloudFormation Vorlage, um einen Linux-basierten Chef 11.10-Stack, -Layer und -Instanz zu erstellen. Die Instance, die durch diese Vorlage erstellt wurde, führt Amazon Linux 2016.03 aus und hat den Instance-Typ `c3.large`.

> **Important**
>
> Die AWS CloudFormation Vorlage muss in derselben Region gespeichert und ausgeführt werden wie der Amazon S3 S3-Bucket, in den Sie später Ihre App hochladen, und in derselben Region, in der Sie später Ihre Pipeline erstellen CodePipeline. CodePipeline Unterstützt derzeit nur den AWS OpsWorks Stacks-Anbieter in der Region USA Ost (Nord-Virginia) (us-east-1). Alle Ressourcen in dieser exemplarischen Vorgehensweise sollten in der Region USA Ost (Nord-Virginia) erstellt werden.
>
> Wenn das Erstellen des Stacks misslingt, haben Sie möglicherweise die maximal zulässige Anzahl der IAM-Rollen für Ihr Konto fast erreicht. Die Stack-Erstellung kann auch fehlschlagen, wenn Ihr Konto Instances des Instance-Typen `c3.large` nicht starten kann. Wenn Sie beispielsweise das AWS kostenlose Kontingent verwenden, erhalten Sie möglicherweise eine Fehlermeldung wie`Root device type: must be included in EBS`. Wenn Ihr Konto Einschränkungen in Bezug auf die Instance-Typen hat, die Sie erstellen dürfen, z. B. Einschränkungen, die durch das AWS kostenlose Kontingent auferlegt werden, versuchen Sie, den Wert des `InstanceType` Parameters im Instance-Block der Vorlage auf einen Instance-Typ zu ändern, den Ihr Konto verwenden kann.

Um einen Stack, eine Ebene und eine Instanz zu erstellen, verwenden Sie AWS CloudFormation

1. Kopieren Sie die folgende AWS CloudFormation Vorlage in ein neues Klartext-Dokument. Speichern Sie die Datei an einem geeigneten Ort auf Ihrem lokalen Computer und geben Sie ihr den Namen NewOpsWorksStack.template oder einen anderen für Sie passenden Namen.

```
{
  "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",
```

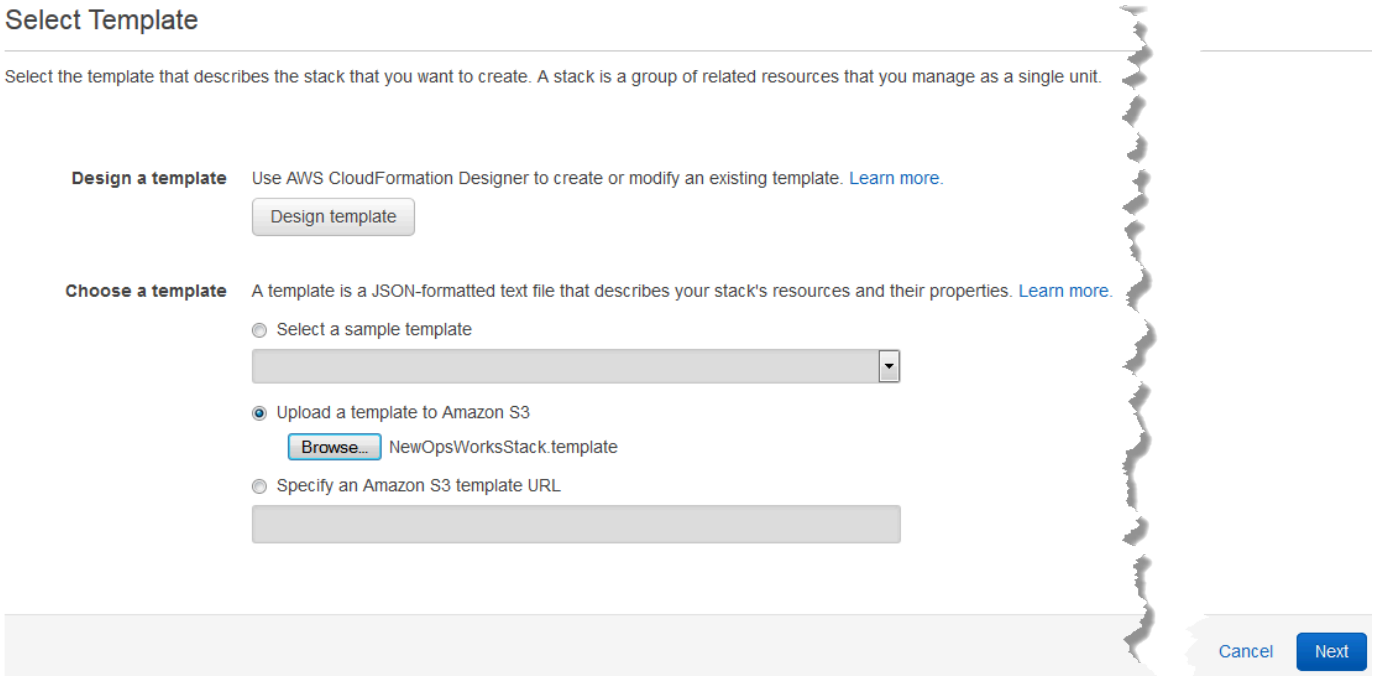
```
        "CPOpsDeploySecGroup"
      ]
    },
    "OpsWorksServiceRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "Service": [
                  {
                    "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. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie auf der AWS CloudFormation Startseite die Option Stack erstellen aus.
4. Wählen Sie auf der Seite Select Template (Vorlage auswählen) im Bereich Choose a template (Eine Vorlage auswählen) die Option Upload a template to Amazon S3 (Eine Vorlage in Amazon S3 hochladen) und anschließend Browse (Durchsuchen) aus.
5. Navigieren Sie zu der AWS CloudFormation Vorlage, die Sie in Schritt 1 gespeichert haben, und wählen Sie dann Öffnen aus. Wählen Sie auf der Seite Vorlage auswählen die Option Weiter aus.



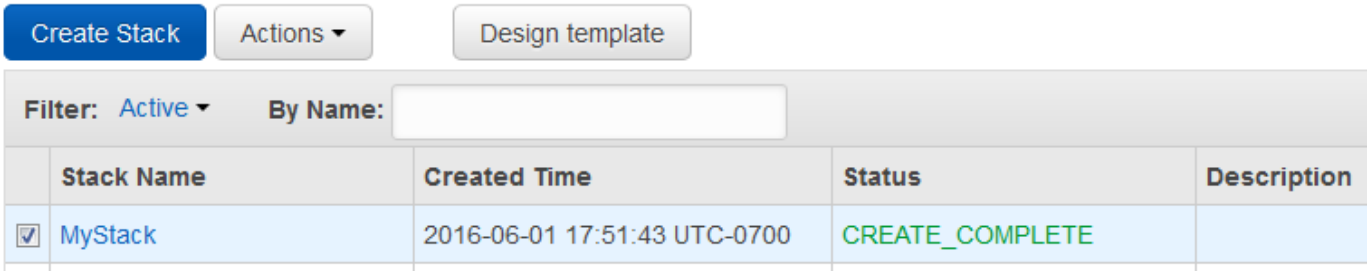
6. Geben Sie auf der Seite „Details angeben“ den Stack oder einen beliebigen Stacknamen MyStack, der für Ihr Konto eindeutig ist, einen Namen. Wenn Sie einen anderen Namen für Ihren Stack auswählen, ändern Sie den Stack-Namen beim Ausführen dieser Anleitung.
7. Geben Sie im Bereich Parameter den Namen eines EC2-Schlüsselpaars ein, das Sie für den Zugriff auf Ihre AWS OpsWorks Stacks-Instance verwenden möchten, nachdem sie erstellt wurde. Wählen Sie Weiter aus.
8. Wählen Sie auf der Seite Optionen Weiter aus. (Einstellungen auf dieser Seite sind für diese Anleitung nicht erforderlich.)

9. Die AWS CloudFormation Vorlage, die Sie in dieser exemplarischen Vorgehensweise verwenden, erstellt IAM-Rollen, ein Instanzprofil und eine Instanz.

   > **Important**
   >
   > Bevor Sie „Erstellen“ wählen, wählen Sie „Kosten“ aus, um die Kosten zu schätzen, die Ihnen durch die Erstellung von AWS Ressourcen mit dieser Vorlage entstehen könnten.

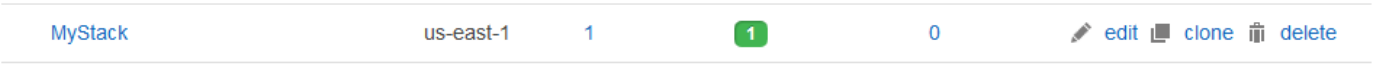
   Wenn das Erstellen von IAM-Ressourcen zulässig ist, aktivieren Sie das Kontrollkästchen Ich bestätige, dass diese Vorlage möglicherweise dazu führt, dass AWS CloudFormation IAM-Ressourcen erstellt, und wählen Sie dann Erstellen aus. Wenn das Erstellen von IAM-Ressourcen nicht akzeptabel ist, können Sie mit diesem Verfahren nicht fortfahren.

10. Auf dem AWS CloudFormation Dashboard können Sie den Fortschritt der Erstellung des Stacks verfolgen. Bevor Sie mit dem nächsten Schritt fortfahren, warten Sie, bis CREATE_COMPLETE in der Spalte Status angezeigt wird.



Um die Stack-Erstellung in AWS OpsWorks Stacks zu überprüfen

1. Öffnen Sie die AWS OpsWorks Konsole unter https://console.aws.amazon.com/opsworks/.
2. Sehen Sie sich im AWS OpsWorks Stacks-Dashboard den Stack an, den Sie erstellt haben.



3. Öffnen Sie den Stack und zeigen Sie den Layer und die Instance an. Beachten Sie, dass die Ebene und die Instanz mit den Namen und anderen Metadaten erstellt wurden, die in der AWS CloudFormation Vorlage angegeben sind. Sie sind bereit, Ihre App in einen Amazon S3 S3-Bucket hochzuladen.

Schritt 2: App-Code in einen Amazon S3 S3-Bucket hochladen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für Neu- als auch für Bestandskunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Da Sie einen Link zu Ihrem Code-Repository als Teil der Pipeline-Einrichtung angeben müssen, halten Sie das Code-Repository bereit, bevor Sie Ihre Pipeline erstellen. In dieser exemplarischen Vorgehensweise laden Sie eine PHP-App in einen Amazon S3 S3-Bucket hoch.

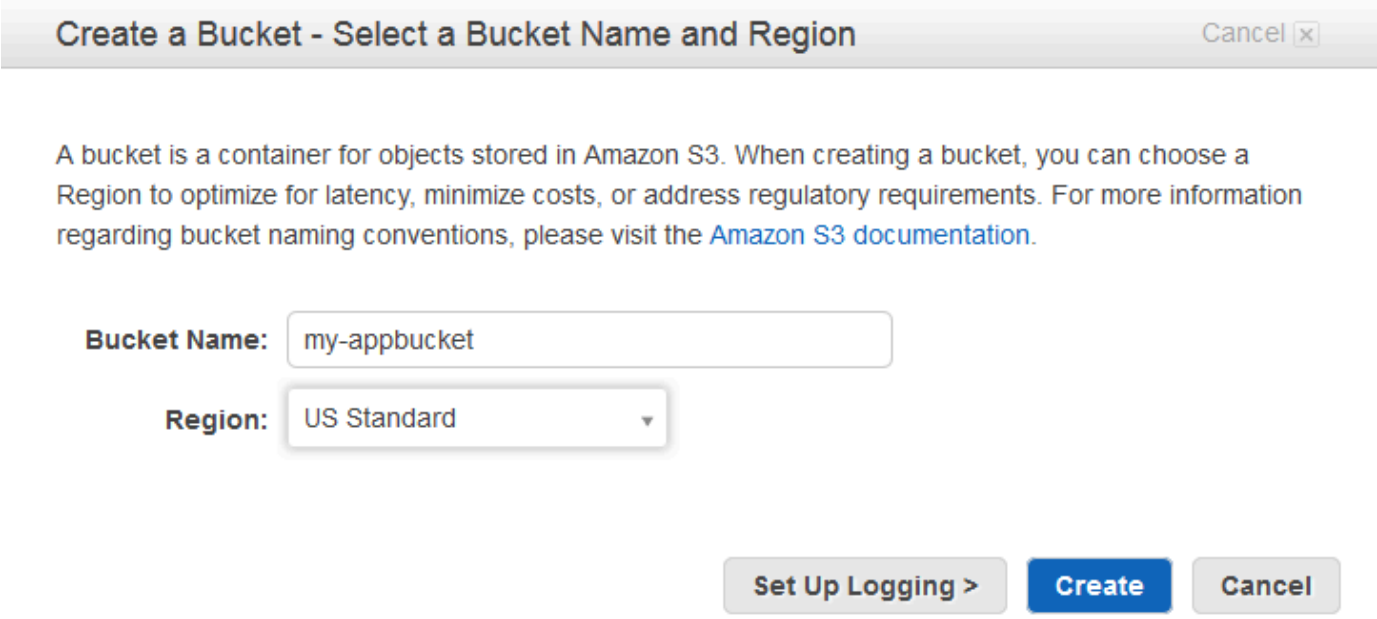
Obwohl Code direkt aus GitHub oder CodeCommit als Quellen verwendet werden CodePipeline kann, zeigt diese exemplarische Vorgehensweise, wie Sie einen Amazon S3 S3-Bucket verwenden. Der Amazon S3 S3-Bucket CodePipeline ermöglicht es, Änderungen am App-Code zu erkennen und die geänderte App automatisch bereitzustellen. Sie können auch einen vorhandenen Bucket verwenden. Stellen Sie sicher, dass der Bucket die in Simple Pipeline Walkthrough (Amazon S3 Bucket) in der CodePipeline Dokumentation beschriebenen Kriterien erfüllt. CodePipeline

> **Important**
>
> Der Amazon S3 S3-Bucket muss sich in derselben Region befinden, in der Sie später Ihre Pipeline erstellen. CodePipeline Unterstützt derzeit nur den AWS OpsWorks Stacks-Anbieter in der Region USA Ost (Nord-Virginia) (us-east-1). Alle Ressourcen in dieser exemplarischen Vorgehensweise sollten in der Region USA Ost (Nord-Virginia) erstellt werden. Der Bucket muss auch versioniert sein, da eine versionierte CodePipeline Quelle erforderlich ist. Weitere Informationen finden Sie unter Verwenden der Versionsverwaltung.

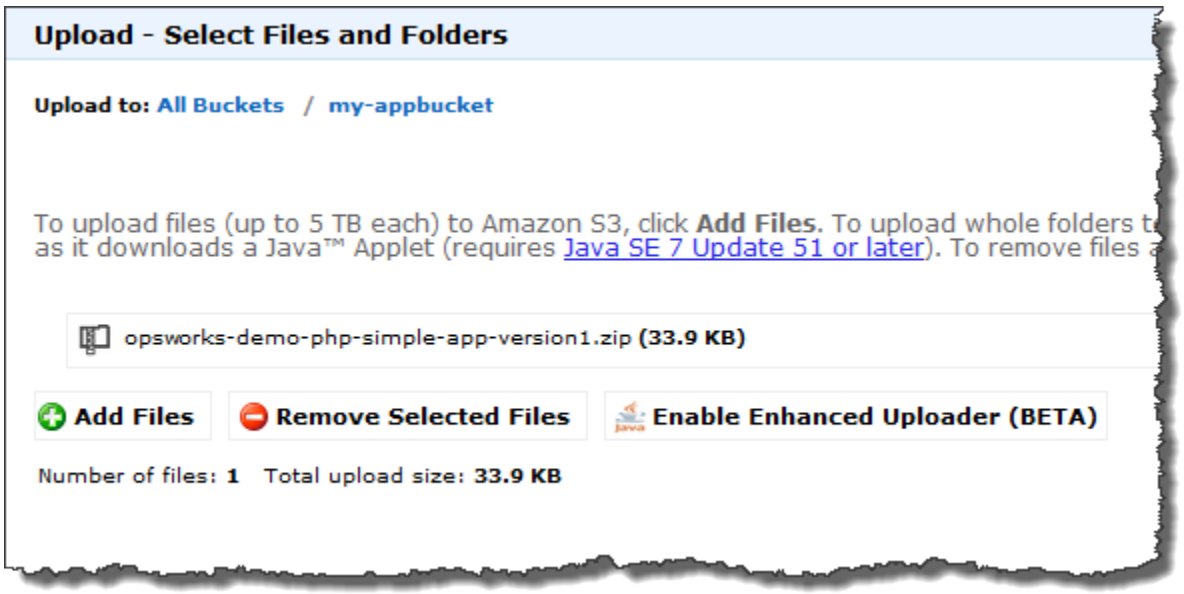
So laden Sie Ihre App in einen Amazon S3 S3-Bucket hoch

1. Laden Sie von der GitHub Website eine ZIP-Datei der AWS OpsWorks Stacks-Beispiel-PHP-Anwendung herunter und speichern Sie sie an einem geeigneten Ort auf Ihrem lokalen Computer.

2. Stellen Sie sicher, dass sich `index.php` und der Ordner `ASSETS` auf der Root-Ebene der heruntergeladenen ZIP-Datei befinden. Wenn dies nicht der Fall ist, extrahieren Sie die Datei und erstellen Sie eine neue ZIP-Datei, die diese Dateien auf der Root-Ebene enthält.
3. Öffnen Sie die Amazon-S3-Konsole unter https://console.aws.amazon.com/s3/.
4. Wählen Sie Create Bucket (Bucket erstellen) aus.
5. Geben Sie auf der Seite Create a Bucket - Select a Bucket Name and Region (Bucket erstellen – Bucket-Namen und Region auswählen) für Bucket Name (Bucket-Name) einen eindeutigen Namen für den Bucket ein. Bucket-Namen müssen für alle AWS Konten eindeutig sein, nicht nur für Ihr eigenes Konto. In dieser Anleitung verwenden wir den Namen **`my-appbucket`**. Sie können als eindeutigen Bucket-Namen aber auch `my-appbucket-`*`yearmonthday`* verwenden. Wählen Sie aus der Dropdown-Liste Region die Option US Standard und anschließend Create (Erstellen) aus. US Standard entspricht `us-east-1`.



6. Wählen Sie den Bucket, den Sie zuvor erstellt haben, aus der Liste All Buckets (Alle Buckets) aus.
7. Wählen Sie auf der Bucket-Seite Upload (Hochladen) aus.
8. Wählen Sie auf der Seite Upload - Select Files and Folders (Hochladen – Dateien und Ordner auswählen) die Option Add Files (Dateien hinzufügen) aus. Suchen Sie nach der ZIP-Datei, die Sie in Schritt 1 gespeichert haben, wählen Sie Open (Öffnen) und anschließend Start Upload (Hochladen starten) aus.



9. Nachdem Sie die Datei hochgeladen haben, wählen Sie die ZIP-Datei aus der Dateiliste in Ihrem Bucket aus und klicken Sie dann auf Properties (Eigenschaften).
10. Kopieren Sie im Bereich Properties (Eigenschaften) den Link zu Ihrer ZIP-Datei und notieren Sie den Link. Sie benötigen den Bucket-Namen und den Teil des Namens der ZIP-Datei dieses Links, um Ihre Pipeline zu erstellen.

Schritt 3: Füge deine App zu AWS OpsWorks Stacks hinzu

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Bevor Sie eine Pipeline erstellen CodePipeline, fügen Sie die PHP-Test-App zu AWS OpsWorks Stacks hinzu. Wenn Sie die Pipeline erstellen, müssen Sie die App auswählen, die Sie zu AWS OpsWorks Stacks hinzugefügt haben.

Halten Sie den Amazon S3 S3-Bucket-Link aus Schritt 10 des vorherigen Verfahrens bereit. Sie benötigen den Link zum Bucket, auf den Sie Ihre Testanwendung gespeichert haben, um diese Anleitung abzuschließen.

Um eine App zu AWS OpsWorks Stacks hinzuzufügen

1. Öffnen MyStackSie in der AWS OpsWorks Stacks-Konsole und wählen Sie im Navigationsbereich Apps aus.
2. Wählen Sie Add app (App hinzufügen) aus.
3. Geben Sie auf der Seite Add App (App hinzufügen) die folgende Information an:

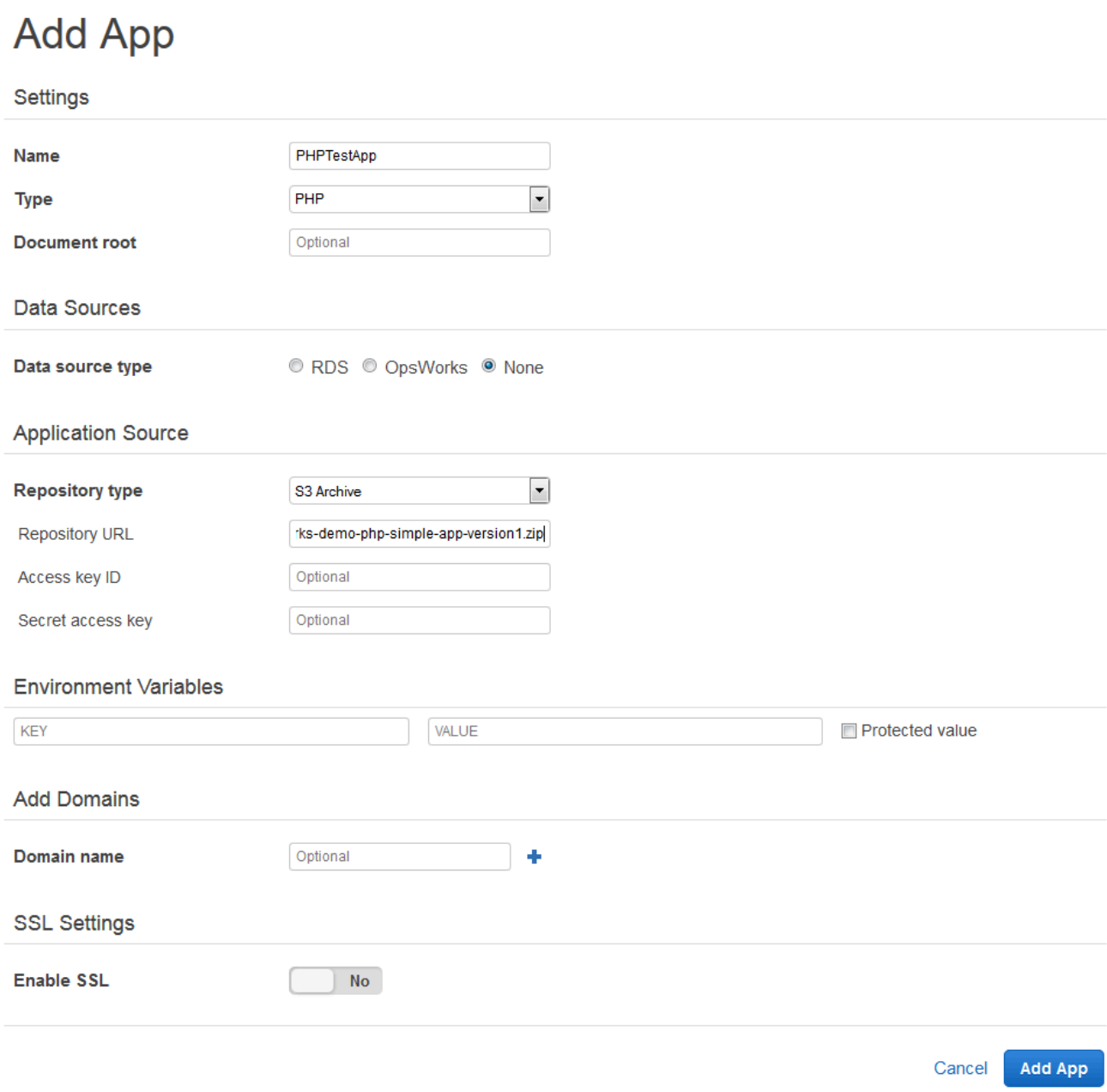
    a. Geben Sie einen Namen für Ihre Anwendung an. In dieser Anleitung wird der Name `PHPTestApp` verwendet.

    b. Wählen Sie in der Dropdown-Liste Type (Typ) die Option PHP aus.

    c. Wählen Sie für Data source type (Datenquellentyp) die Option None (Kein) aus. Diese Anwendung erfordert keine externe Datenbank oder Datenquelle.

    d. Wählen Sie aus der Dropdown-Liste Repository type (Repository-Typ) die Option S3 Archive (S3-Archiv) aus.

    e. Fügen Sie in das Textfeld Repository URL (Repository-URL) die URL ein, die Sie in Schritt 10 von Schritt 2: App-Code in einen Amazon S3 S3-Bucket hochladen kopiert haben. Ihr Formular sollte ähnlich wie folgt aussehen:



4. Sie müssen in diesem Formular keine weiteren Einstellungen ändern. Wählen Sie Add App (Anwendung hinzufügen) aus.
5. Wenn die TestAppPHP-App in der Liste auf der Apps-Seite angezeigt wird, fahren Sie mit dem nächsten Verfahren fort. Schritt 4: Erstellen Sie eine Pipeline in CodePipeline

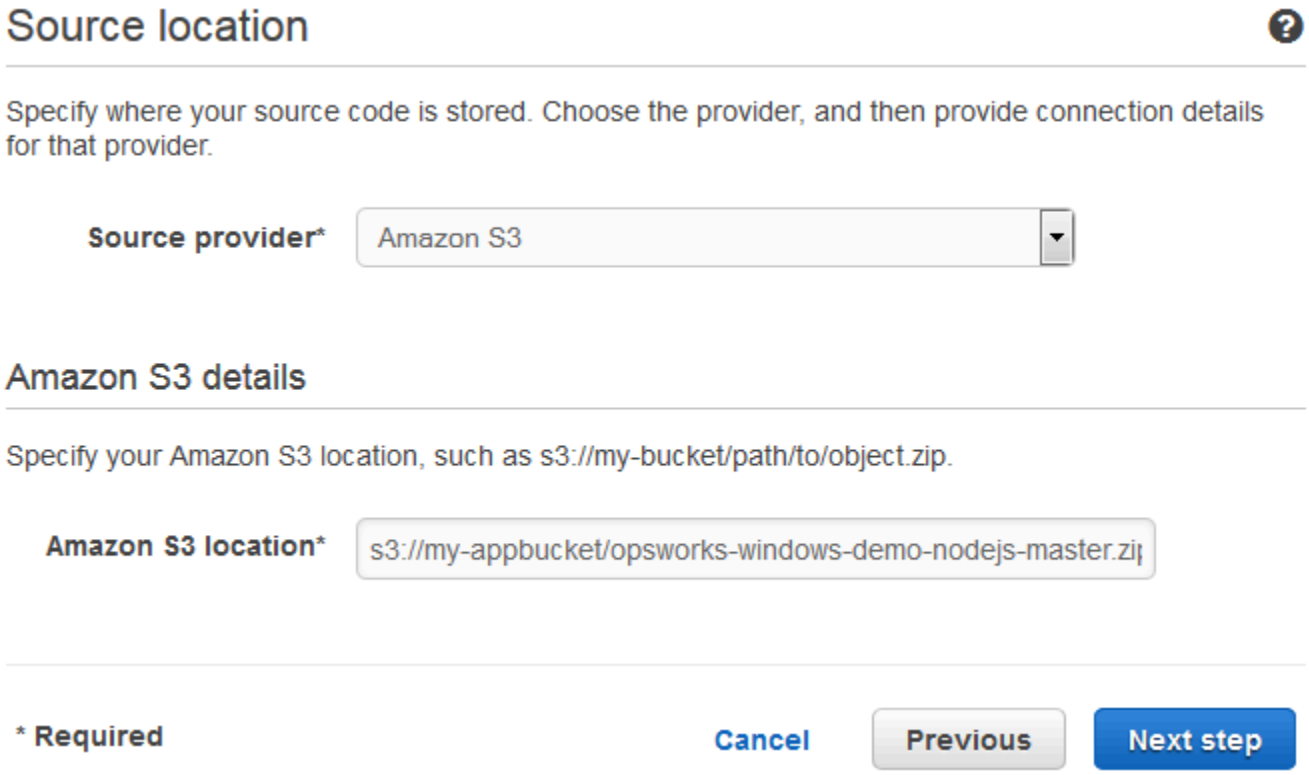
Schritt 4: Erstellen Sie eine Pipeline in CodePipeline

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

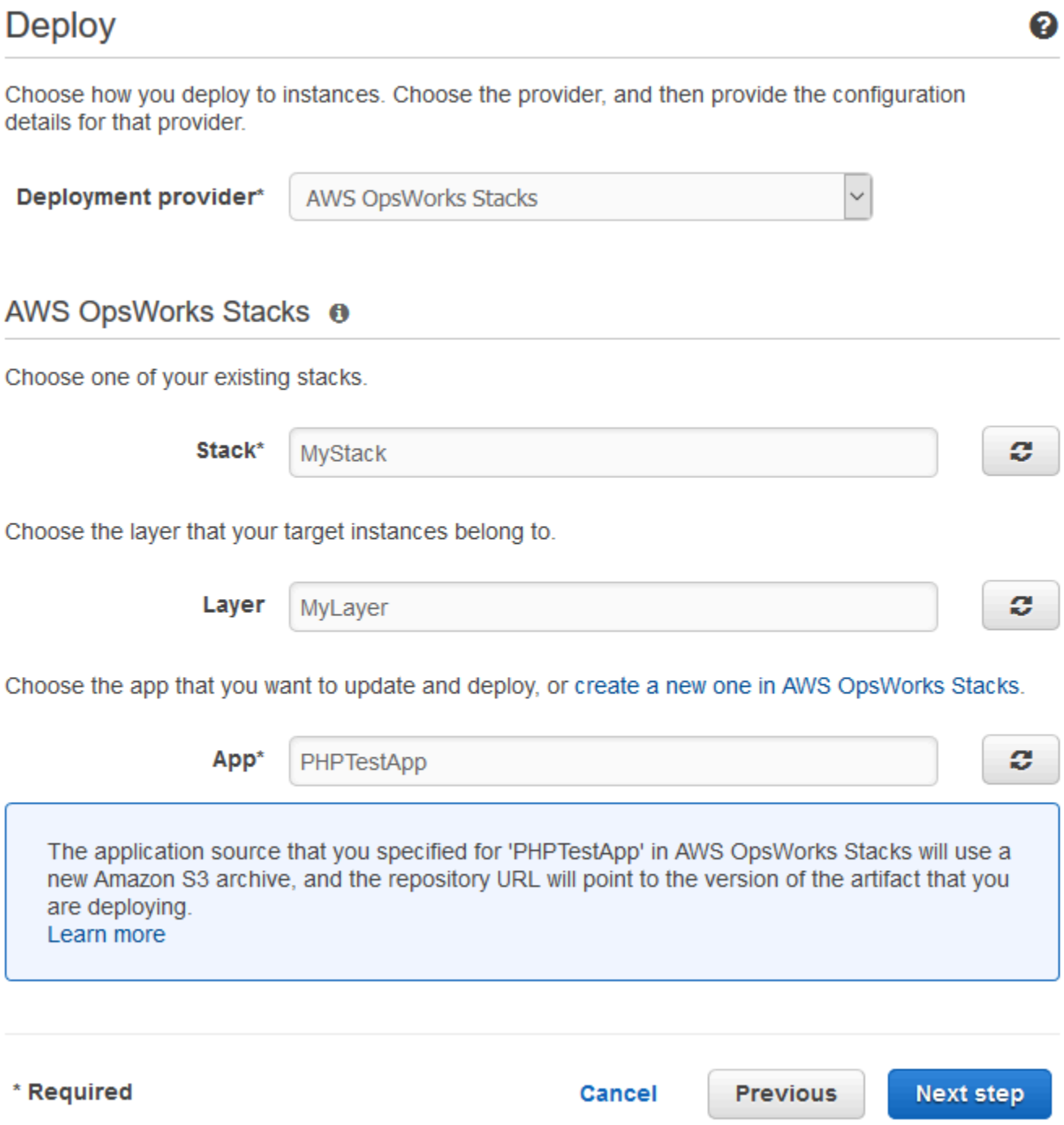
Nachdem Sie einen Stack mit einer Ebene und mindestens einer Instanz in AWS OpsWorks Stacks konfiguriert haben, erstellen Sie eine Pipeline CodePipeline mit AWS OpsWorks Stacks als Anbieter, um Apps oder Chef-Kochbücher für Ihre Stacks-Ressourcen bereitzustellen. AWS OpsWorks

So erstellen Sie eine Pipeline

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/codepipeline/ CodePipeline .
2. Wählen Sie Create pipeline (Pipeline erstellen) aus.
3. Geben **`MyOpsWorksPipeline`** Sie auf der CodePipeline Seite Erste Schritte mit einen beliebigen anderen Pipelinenamen ein, der für Ihr Konto eindeutig ist, und wählen Sie dann Weiter aus.
4. Wählen Sie auf der Seite Source Location (Quellspeicherort) die Option Amazon S3 aus der Dropdown-Liste Source provider (Quellanbieter) aus.
5. Geben Sie im Bereich Amazon S3 S3-Details Ihren Amazon S3 S3-Bucket-Pfad im folgenden Format ein**`s3://`*`bucket-name`*`/`*`file name`***. Verwenden Sie dabei den Link, den Sie in Schritt 10 von Schritt 2: App-Code in einen Amazon S3 S3-Bucket hochladen notiert haben. In dieser Anleitung ist der Pfad `s3://my-appbucket/opsworks-demo-php-simple-app-version1.zip`. Klicken Sie auf Nächster Schritt.



6. Wählen Sie auf der Seite Build die Option No Build (Kein Build) aus der Dropdown-Liste und anschließend Next step (Nächster Schritt) aus.
7. Wählen Sie auf der Seite Deploy (Bereitstellen) als Bereitstellungsanbieter AWS OpsWorks Stacks aus.



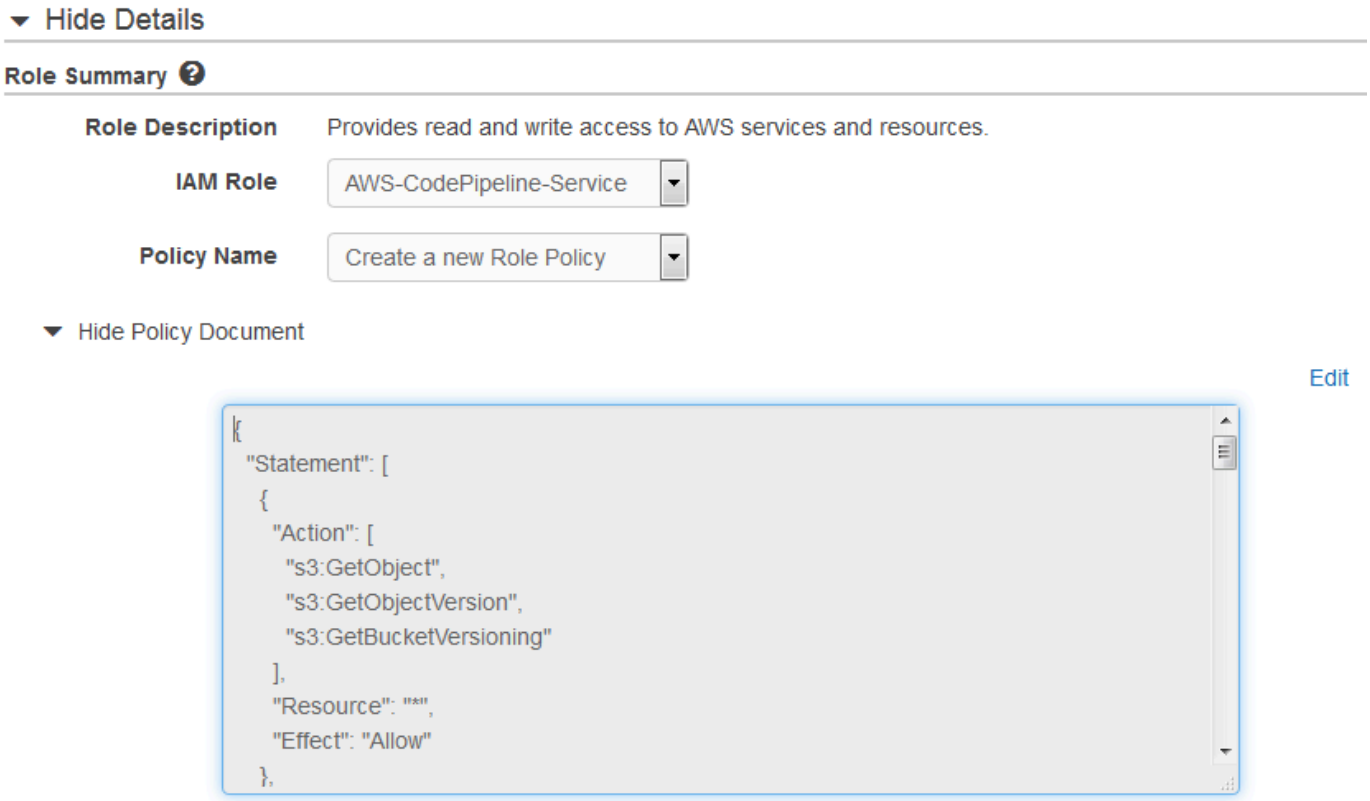
8. Geben Sie im Feld Stack `MyStack` oder den Namen des Stacks ein, den Sie in Schritt 1: Erstellen eines Stacks, Layers und einer Instance in AWS OpsWorks Stacks erstellt haben.
9. Geben Sie im Feld Layer `MyLayer` oder den Namen des Layers ein, den Sie in Schritt 1: Erstellen eines Stacks, Layers und einer Instance in AWS OpsWorks Stacks erstellt haben.

10. Wählen Sie im Feld App die App aus, in der Sie auf Amazon S3 hochgeladen habenSchritt 2: App-Code in einen Amazon S3 S3-Bucket hochladen, und wählen Sie dann Nächster Schritt aus.
11. Wählen Sie auf der Seite AWS Service Role (AWS-Servicerolle) Create Role (Rolle erstellen) aus.

    Es öffnet sich ein neues Fenster mit einer IAM-Konsolenseite, auf der die Rolle beschrieben wird, die für Sie erstellt wird. `AWS-CodePipeline-Service` Wählen Sie aus der Dropdown-Liste Policy name (Richtlinienname) die Option Create new policy (Neue Richtlinie erstellen) aus. Stellen Sie sicher, dass das Richtliniendokument den folgenden Inhalt hat. Wählen Sie Edit (Bearbeiten) aus, um gegebenenfalls das Richtliniendokument zu ändern.

    ```
    {
        "Statement": [
            {
                "Action": [
                    "s3:GetObject",
                    "s3:GetObjectVersion",
                    "s3:GetBucketVersioning"
                ],
                "Resource": "*",
                "Effect": "Allow"
            },
            {
                "Action": "opsworks:*",
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    }
    ```

    Wenn Sie alle gewünschten Änderungen für das Richtliniendokument ausgeführt haben, wählen Sie Allow (Zulassen) aus. Ihre Änderungen werden in der IAM-Konsole angezeigt.



> **Note**
>
> Wenn die Rollenerstellung fehlschlägt, liegt das möglicherweise daran, dass Sie bereits über eine IAM-Rolle mit dem Namen CodePipelineAWS-Service verfügen. Wenn Sie die Rolle AWS- CodePipeline -Service vor Mai 2016 verwendet haben, verfügt die Rolle möglicherweise nicht über die Berechtigungen, AWS OpsWorks Stacks als Bereitstellungsanbieter zu verwenden. In diesem Fall müssen Sie die Richtlinienerklärung wie in diesem Schritt beschrieben aktualisieren. Wenn Ihnen eine Fehlermeldung angezeigt wird, kehren Sie zum Anfang dieses Schritts zurück und wählen anstelle von Create role (Rolle erstellen) die Option Use existing role (Vorhandene Rolle verwenden) aus. Wenn Sie eine vorhandene Rolle verwenden, sollte die Rolle einer Richtlinie zugewiesen sein, die die Berechtigungen, wie in diesem Schritt dargestellt, enthält. Weitere Informationen zur Servicerolle und deren Richtlinienanweisung finden Sie unter Bearbeiten einer Richtlinie für eine IAM-Servicerolle.

12. Wenn die Rollenerstellung erfolgreich ist, wird die IAM-Seite geschlossen und Sie kehren zur Seite AWS-Servicerolle zurück. Klicken Sie auf Nächster Schritt.

13. Überprüfen Sie Ihre Auswahl auf der Seite Review your pipeline (Ihre Pipeline überprüfen) und wählen Sie dann Create pipeline (Pipeline erstellen) aus.

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. Wenn Ihre Pipeline bereit ist, sollte sie automatisch damit beginnen. Ihren Quellcode zu ermitteln und Ihre Anwendung zu Ihrem Stack bereitzustellen. Dieser Vorgang kann einige Minuten dauern.

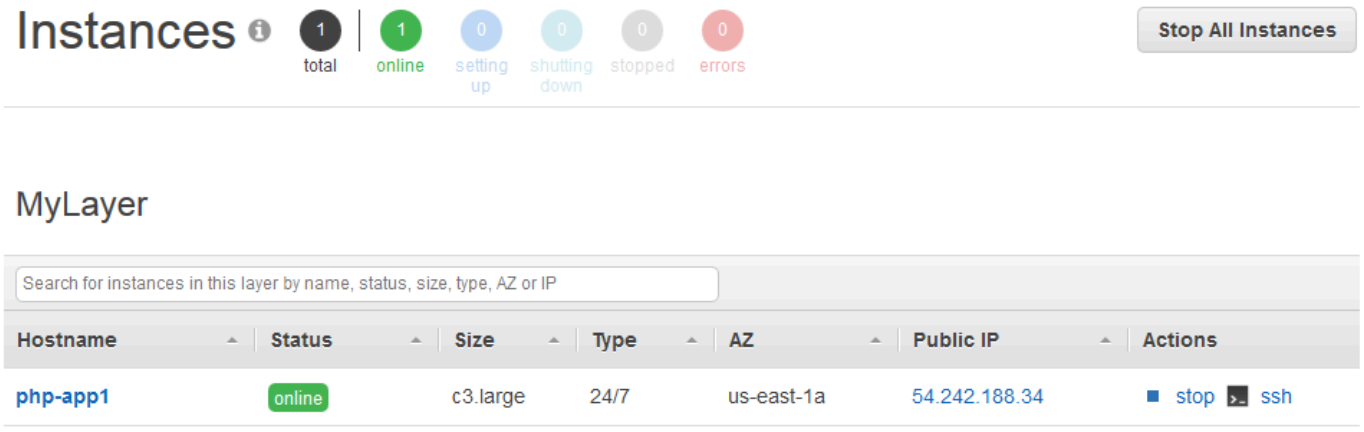
Schritt 5: Überprüfen der App-Bereitstellung in Stacks AWS OpsWorks

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

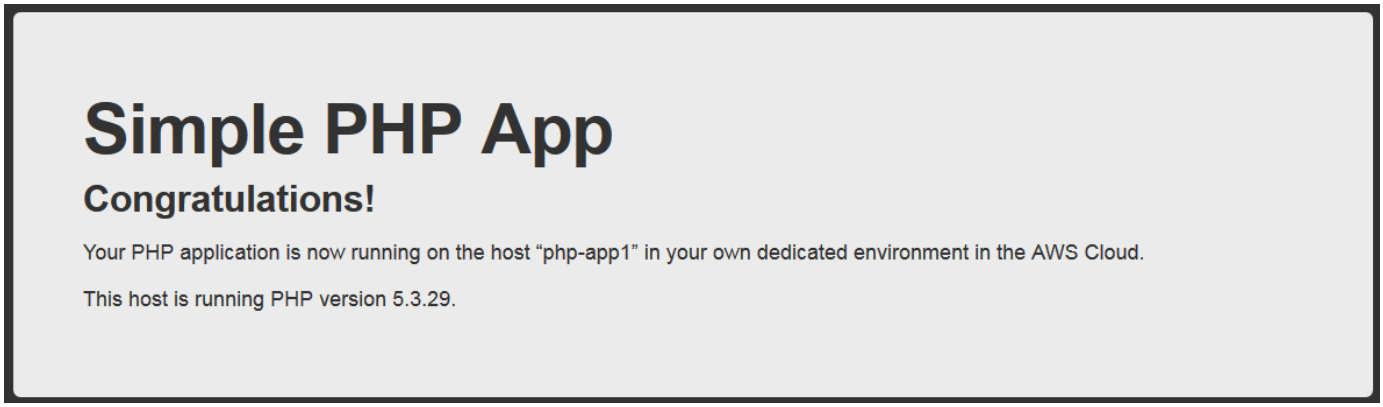
Um zu überprüfen, ob die PHP-App in Ihrem Stack CodePipeline bereitgestellt wurde, melden Sie sich bei der Instanz an, in der Sie sie erstellt haben. Schritt 1: Erstellen eines Stacks, Layers und einer Instance in AWS OpsWorks Stacks Sie sollten die PHP-Webanwendung sehen und nutzen können.

Um die App-Bereitstellung in Ihrer AWS OpsWorks Stacks-Instanz zu überprüfen

1. Öffnen Sie die AWS OpsWorks Konsole unter https://console.aws.amazon.com/opsworks/.
2. Wählen Sie im AWS OpsWorks Stacks-Dashboard und wählen MyStackSie MyLayerdann.
3. Wählen Sie im Navigationsbereich Instances und anschließend die öffentliche IP-Adresse der Instance aus, die Sie erstellt haben, um die Webanwendung anzuzeigen.



Die Anwendung wird auf einer neuen Registerkarte angezeigt werden.



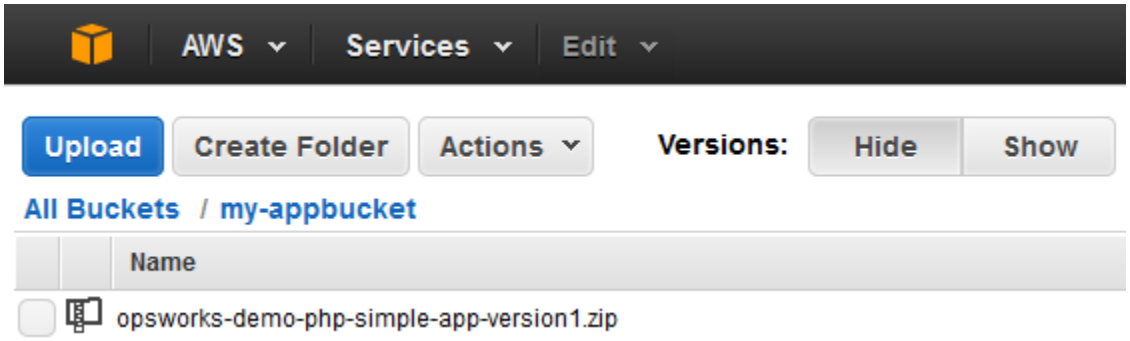
Schritt 6 (optional): Aktualisieren Sie den App-Code, um zu sehen, wie Ihre App CodePipeline automatisch erneut bereitgestellt wird

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn Sie Änderungen am Code in Apps oder Cookbooks vornehmen, die Sie mithilfe von Using bereitgestellt haben CodePipeline, werden die aktualisierten Artefakte automatisch CodePipeline auf Ihren Zielinstanzen (in diesem Fall auf einem AWS OpsWorks Ziel-Stacks-Stack) bereitgestellt. In diesem Abschnitt wird beschrieben, wie die Anwendung automatisch erneut bereitgestellt wird, wenn Sie den Code in Ihrer PHP-Beispielanwendung aktualisieren.

So bearbeiten Sie den Code in der Beispielanwendung

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/s3/.
2. Öffnen Sie den Bucket, in dem Sie Ihre PHP-Beispielanwendung speichern.



3. Wählen Sie die ZIP-Datei, die die Anwendung enthält. Wählen Sie im Menü Actions die Option Download aus.
4. Öffnen Sie im Dialogfeld mit der rechten Maustaste das Kontextmenü, wählen Sie Download (Herunterladen) aus und speichern Sie dann die ZIP-Datei an einem geeigneten Ort. Wählen Sie OK aus.
5. Extrahieren Sie die Inhalte der ZIP-Datei an einem geeigneten Ort. Möglicherweise müssen Sie Berechtigungen für die extrahierten Ordner und deren Unterordner und Inhalte ändern, sodass eine Bearbeitung zugelassen wird. Öffnen Sie im Ordner `opsworks-demo-php-simple-app-version1` die Datei `index.php`, um sie zu bearbeiten.
6. Suchen Sie nach der Zeichenfolge `Your PHP application is now running`. Ersetzen Sie den Text `Your PHP application is now running` durch `You've just deployed your first app to AWS OpsWorks with AWS CodePipeline,`. Bearbeiten Sie nicht die Variablen.

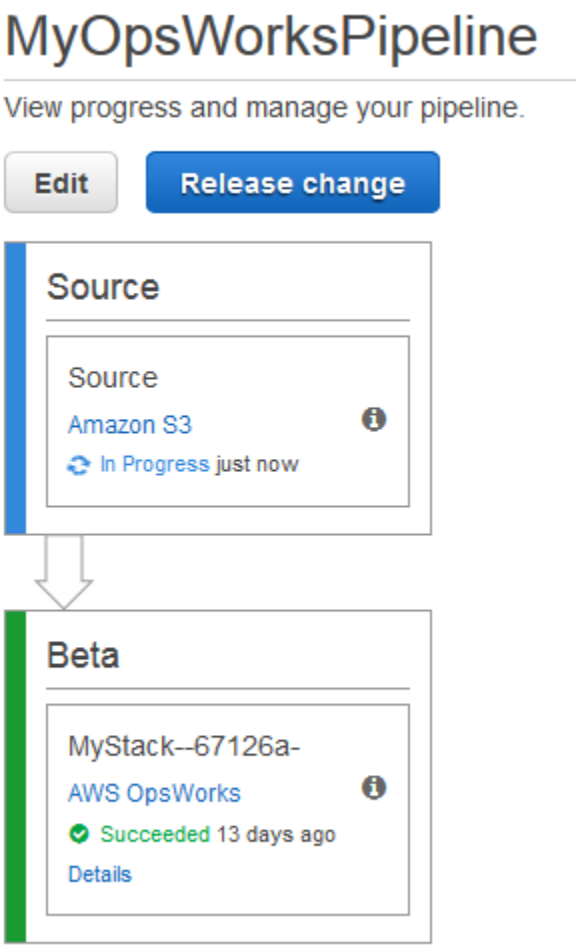
```
<body>
    <div class="container">
        <div class="hero-unit">
            <h1>Simple PHP App</h1>
            <h2>Congratulations!</h2>
            <p>You've just deployed your first app to AWS OpsWorks with AWS CodePipeline,</p>
            <p>on the host &ldquo;<?php echo gethostname(); ?>&rdquo; </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>
```

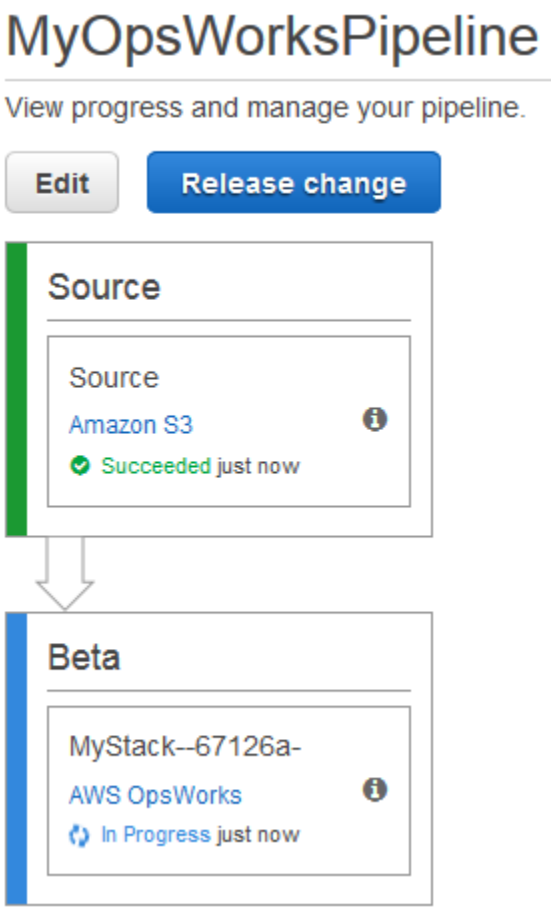
7. Speichern und schließen Sie die Datei `index.php`.
8. Packen Sie das Verzeichnis `opsworks-demo-php-simple-app-version1` und speichern Sie die ZIP-Datei. Ändern Sie nicht den Namen der ZIP-Datei.

9. Laden Sie die neue ZIP-Datei in Ihren Amazon S3 S3-Bucket hoch. In dieser Anleitung ist der Name des Buckets `my-appbucket`.
10. Öffnen Sie die CodePipeline Konsole und öffnen Sie Ihre AWS OpsWorks Stacks-Pipeline (MyOpsWorksPipeline). Wählen Sie Release Change (Versionsänderung) aus.

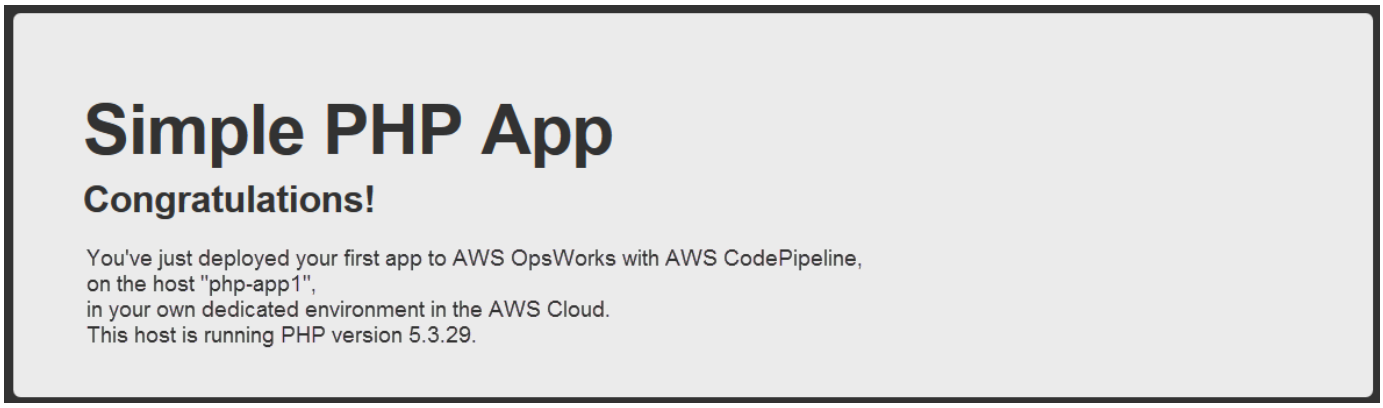
    (Sie können warten CodePipeline , bis Sie die Codeänderung aus der aktualisierten Version der App in Ihrem Amazon S3 S3-Bucket feststellen. Um Ihnen Zeit zu sparen, werden Sie in dieser exemplarischen Vorgehensweise aufgefordert, einfach Release Change auszuwählen.)
11. Beobachten Sie, CodePipeline wie die einzelnen Phasen der Pipeline durchlaufen werden. CodePipeline Erkennt zunächst Änderungen am Quellartefakt.



CodePipeline verschiebt den aktualisierten Code auf Ihren Stack in AWS OpsWorks Stacks.



12. Wenn beide Phasen der Pipeline erfolgreich abgeschlossen wurden, öffnen Sie Ihren Stack in AWS OpsWorks Stacks (). MyStack
13. Wählen Sie auf der MyStackEigenschaftenseite Instances aus.
14. Wählen Sie in der Spalte Public IP (Öffentliche IP-Adresse) die öffentliche IP-Adresse Ihrer Instance aus, um den Text der aktualisierten Anwendung anzuzeigen.



Schritt 7 (optional): Bereinigen der Ressourcen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Um zu verhindern, dass Ihr AWS-Konto ungewollt belastet wird, können Sie die AWS Ressourcen löschen, die Sie für diese exemplarische Vorgehensweise verwendet haben. Zu diesen AWS Ressourcen gehören der AWS OpsWorks Stacks-Stack, die IAM-Rolle und das Instance-Profil sowie die Pipeline, in der Sie sie erstellt haben. CodePipeline Möglicherweise möchten Sie diese AWS Ressourcen jedoch weiterhin verwenden, wenn Sie mehr über AWS OpsWorks Stacks und erfahren. CodePipeline Wenn Sie diese Ressourcen behalten möchten, haben Sie diese Anleitung abgeschlossen.

So löschen Sie die Anwendung aus dem Stack

Da Sie die App nicht als Teil Ihrer AWS CloudFormation Vorlage erstellt oder angewendet haben, löschen Sie die PHP-Test-App, bevor Sie den Stack in AWS CloudFormation löschen.

1. Wählen Sie in der AWS OpsWorks Stacks-Konsole im Navigationsbereich des Dienstes Apps aus.
2. Wählen Sie auf der Apps-Seite PHP TestApp und dann unter Aktionen die Option Löschen aus. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Löschen. AWS OpsWorks Stacks löscht die App.

So löschen Sie den Stack

Da Sie den Stack erstellt haben, indem Sie eine AWS CloudFormation Vorlage ausgeführt haben, können Sie den Stack, einschließlich der Ebene, der Instanz, des Instanzprofils und der Sicherheitsgruppe, die die Vorlage erstellt hat, in der AWS CloudFormation Konsole löschen.

1. Öffnen Sie die AWS CloudFormation Konsole.

2. Wählen Sie im AWS CloudFormation Konsolen-Dashboard den Stack aus, den Sie erstellt haben (MyStack). Wählen Sie im Menü Actions (Aktionen) die Option Delete Stack (Stack löschen) aus. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Yes, Delete (Ja, löschen) aus.
3. Warten Sie, bis DELETE_COMPLETE in der Spalte Status (Status) für den Stack angezeigt wird.

So löschen Sie die Pipeline

1. Öffnen Sie die CodePipeline Konsole.
2. Wählen Sie im CodePipeline Dashboard die Pipeline aus, die Sie für diese exemplarische Vorgehensweise erstellt haben.
3. Wählen Sie auf der Pipeline-Seite Edit (Bearbeiten) aus.
4. Klicken Sie auf der Seite Edit auf Delete. Wenn Sie aufgefordert werden, Ihre Entscheidung zu bestätigen, wählen Sie Delete aus.

Verwenden der AWS OpsWorks Stacks-CLI

Important

Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Die AWS OpsWorks Stacks-Befehlszeilenschnittstelle (CLI) bietet die gleiche Funktionalität wie die Konsole und kann für eine Vielzahl von Aufgaben verwendet werden. Die AWS OpsWorks Stacks-CLI ist Teil der AWS CLI. Weitere Informationen, einschließlich der Installation und Konfiguration von AWS CLI, finden Sie unter Was ist die AWS-Befehlszeilenschnittstelle? . Eine vollständige Beschreibung für jeden Befehl finden Sie in der AWS OpsWorks Stacks-Referenz.

Note

Wenn Sie eine Windows-basierte Workstation verwenden, können Sie auch die AWS Tools für Windows ausführen, PowerShell um AWS OpsWorks Stacks-Operationen von

der Befehlszeile aus auszuführen. Weitere Informationen finden Sie unter AWS-Tools für Windows PowerShell.

AWS OpsWorks Stacks-Befehle haben das folgende allgemeine Format:

```
aws opsworks --region us-west-1 opsworks command-name [--argument1 value] [...]
```

Wenn ein Argumentwert ein JSON-Objekt ist, sollten Sie die "-Zeichen mit Escape-Zeichen schützen, andernfalls kann der Befehl zu einer Fehlermeldung führen, die besagt das JSON ungültig ist. Wenn beispielsweise das JSON-Objekt `"{"somekey":"somevalue"}"` lautet, formatieren Sie es als `"{\"somekey\":\"somevalue\"}"`. Alternativ speichern Sie das JSON-Objekt in eine Datei und verwenden Sie `file://`, um es in die Befehlszeile einzufügen. Im folgenden Beispiel wird eine Anwendung mit einem Anwendungsquellobjekt erstellt, das in appsource.json gespeichert ist.

```
aws opsworks --region us-west-1 create-app --stack-id 8c428b08-a1a1-46ce-a5f8-
feddc43771b8 --name SimpleJSP --type java --app-source file://appsource.json
```

Die meisten Befehle geben einen oder mehrere Werte als JSON-Objekt verpackt zurück. Die folgenden Abschnitte enthalten einige Beispiele. Eine detaillierte Beschreibung der Rückgabewerte für die einzelnen Befehle finden Sie in der AWS OpsWorks Stacks-Referenz.

> **Note**
>
> AWS CLI Befehle müssen eine Region angeben, wie in den Beispielen gezeigt. In der folgenden Tabelle sind gültige Werte für den --region-Parameter aufgeführt. Um Ihre AWS OpsWorks Stacks-Befehlszeichenfolgen zu vereinfachen, konfigurieren Sie die CLI so, dass sie Ihre Standardregion angibt, sodass Sie den `--region` Parameter weglassen können. Wenn Sie normalerweise an mehreren regionalen Endpunkten arbeiten, konfigurieren Sie den nicht so, dass er einen regionalen AWS CLI Standardendpunkt verwendet. Der Endpunkt für die Region Kanada (Mitte) ist AWS CLI nur in der API verfügbar. Er ist nicht für Stacks verfügbar, die Sie in der erstellen. AWS Management Console Weitere Informationen finden Sie unter Konfigurieren der AWS-Region.
>
> | Name der Region | Befehlscode |
> | --- | --- |
> | Region USA Ost (Ohio) | us-east-2 |

Name der Region	Befehlscode
Region USA Ost (Nord-Virginia)	us-east-1
Region US West (N. California)	us-west-1
Region USA West (Oregon)	us-west-2
Region Kanada (Zentral)	ca-central-1
Region Europa (Irland)	eu-west-1
Region Europa (London)	eu-west-2
Region Europa (Paris)	eu-west-3
Region Europa (Frankfurt)	eu-central-1
Region Asien-Pazifik (Tokio)	ap-northeast-1
Region Asien-Pazifik (Seoul)	ap-northeast-2
Region Asien-Pazifik (Mumbai)	ap-south-1
Region Asien-Pazifik (Singapur)	ap-southeast-1
Region Asien-Pazifik (Sydney)	ap-southeast-2
Region Südamerika (São Paulo)	sa-east-1

Um einen CLI-Befehl ausführen zu können, müssen Sie über die entsprechenden Berechtigungen verfügen. Weitere Informationen zu AWS OpsWorks Stacks-Berechtigungen finden Sie unter Verwalten von Benutzerberechtigungen. Um die erforderlichen Berechtigungen für einen bestimmten Befehl zu ermitteln, sehen Sie sich die Referenzseite des Befehls in der AWS OpsWorks Stacks-Referenz an.

In den folgenden Abschnitten wird beschrieben, wie Sie die AWS OpsWorks Stacks-CLI verwenden, um eine Vielzahl allgemeiner Aufgaben auszuführen.

Erstellen einer Instance (create-Instance)

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Verwenden Sie zum Erstellen einer Instance auf einem bestimmten Stack den Befehl create-instance.

Themen

- Erstellen einer Instance mit einem standardmäßigen Hostnamen
- Erstellen einer Instance mit einem themenbezogenen Hostnamen
- Erstellen einer Instance mit einem benutzerdefinierten AMI

Erstellen einer Instance mit einem standardmäßigen Hostnamen

```
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"
```

Die Argumente lauten wie folgt:

- `stack-id`— Sie können die Stack-ID auf der Einstellungsseite des Stacks auf der Konsole abrufen (suchen Sie nach der OpsWorks ID) oder indem Sie describe-stacks aufrufen.
- `layer-ids`— Sie können Layer-IDs von der Detailseite der Ebene auf der Konsole abrufen (suchen Sie nach der OpsWorks ID) oder indem Sie describe-layers aufrufen. In diesem Beispiel ist die Instance nur einem Layer zugehörig.
- `instance-type` - Die Spezifikation, mit der Arbeitsspeicher, CPU, Speicherkapazität und Stundensatz für die Instance bestimmt wird; in diesem Beispiel `m1.large`.
- `os` - Das Instance-Betriebssystem; in diesem Beispiel Amazon Linux.

Der Befehl gibt ein JSON-Objekt mit der Instance-ID zurück:

```
{
    "InstanceId": "5f9adeaa-c94c-42c6-aeef-28a5376002cd"
}
```

In diesem Beispiel wird eine Instance mit einem standardmäßigen Hostnamen erstellt, der einfach eine Ganzzahl ist. Im folgenden Abschnitt wird beschrieben, wie Sie eine Instance mit einem auf Grundlage eines Themas erzeugten Hostnamen erstellen.

Erstellen einer Instance mit einem themenbezogenen Hostnamen

Darüber hinaus können Sie eine Instance mit einem themenbezogenen Hostnamen erstellen. Beim Erstellen des Stacks geben Sie das Thema an. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks .Um die Instanz zu erstellen, rufen Sie zuerst auf, um einen Namen get-hostname-suggestionzu generieren. Beispielsweise:

```
C:\>aws opsworks get-hostname-suggestion --region us-west-1 --layer-id 5c8c272a-
f2d5-42e3-8245-5bf3927cb65b
```

Wenn Sie das standardmäßige `Layer Dependent`-Thema angeben, fügt `get-hostname-suggestion` einfach eine Ziffer zum Kurznamen des Layers hinzu. Weitere Informationen finden Sie unter Erstellen eines neuen Stacks.

Der Befehl gibt den erzeugten Hostnamen zurück.

```
{
    "Hostname": "php-app2",
    "LayerId": "5c8c272a-f2d5-42e3-8245-5bf3927cb65b"
}
```

Anschließend können Sie das `hostname`-Argument verwenden, um den erzeugten Namen an `create-instance` zu übermitteln:

```
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"
```

Erstellen einer Instance mit einem benutzerdefinierten AMI

Der nachstehende Befehl create-instance erstellt eine Instance mit einem benutzerdefinierten AMI, das aus der Stack-Region stammen muss. Weitere Informationen zum Erstellen eines benutzerdefinierten AMI für AWS OpsWorks Stacks finden Sie unterVerwenden von benutzerdefinierten 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
```

Die Argumente lauten wie folgt:

- `stack-id`— Sie können die Stack-ID von der Einstellungsseite des Stacks auf der Konsole abrufen (suchen Sie nach der OpsWorks ID) oder indem Sie describe-stacks aufrufen.
- `layer-ids`— Sie können Layer-IDs von der Detailseite der Ebene auf der Konsole abrufen (suchen Sie nach der OpsWorks ID) oder indem Sie describe-layers aufrufen. In diesem Beispiel ist die Instance nur einem Layer zugehörig.
- `instance-type` – Der Wert definiert Arbeitsspeicher, CPU, Speicherkapazität und Stundenrate der Instance und muss mit dem AMI kompatibel sein (in diesem Beispiel `c3.large`).
- `os` – Das Betriebssystem der Instance, das für ein benutzerdefiniertes AMI auf `Custom` gesetzt sein muss.
- `ami-id` - Die AMI-ID, beispielsweise `ami-6c61f104`.

> **Note**
>
> Wenn Sie ein benutzerdefiniertes AMI verwenden, werden Block-Gerät-Zuweisungen nicht unterstützt, und die Werte, die Sie für die Option `--block-device-mappings` angeben, werden ignoriert.

Der Befehl gibt ein JSON-Objekt mit der Instance-ID zurück:

```
{
    "InstanceId": "5f9adeaa-c94c-42c6-aeef-28a5376002cd"
}
```

Bereitstellen einer Anwendung (create-deployment)

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Verwenden Sie den Befehl create-deployment, um die Bereitstellung einer Anwendung an einem bestimmten Stack anzuweisen.

Themen

- Bereitstellen einer Anwendung

Bereitstellen einer Anwendung

```
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\"}"
```

Die Argumente lauten wie folgt:

- `stack-id`— Sie können die Stack-ID auf der Einstellungsseite des Stacks auf der Konsole (suchen Sie nach der OpsWorks ID) oder telefonisch abrufen. `describe-stacks`
- `app-id`— Sie können die App-ID auf der Detailseite der App abrufen (suchen Sie nach der OpsWorks ID) oder indem Sie describe-apps aufrufen.

- `command` - Das Argument übernimmt ein JSON-Objekt, das den Befehlsnamen auf `deploy` setzt, welcher die angegebene Anwendung an den Stack bereitstellt.

  Beachten Sie, dass alle "-Zeichen im JSON-Objekt mit Escape-Zeichen versehen sind. Andernfalls gibt der Befehl möglicherweise eine Fehlermeldung aus, die besagt, dass JSON ungültig ist.

Der Befehl gibt ein JSON-Objekt mit der Bereitstellungs-ID zurück:

```
{
    "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"
}
```

> **Note**
>
> Im vorherigen Beispiel erfolgt eine Bereitstellung an jede Instance im Stack. Um eine angegebene Teilmenge von Instances bereitzustellen, fügen Sie ein Argument `instance-ids` hinzu und listen Sie die Instance-ID auf.

Auflisten der Anwendungen eines Stacks (describe-Apps)

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Verwenden Sie den Befehl describe-apps, um die Anwendungen eines Stacks aufzulisten oder detaillierte Informationen über die angegebenen Anwendungen zu erhalten.

```
aws opsworks --region us-west-1 describe-apps --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

Das vorherige Beispiel gibt ein JSON-Objekt mit Informationen über jede Anwendung zurück. In diesem Beispiel ist nur eine Anwendung vorhanden. Eine Beschreibung aller Parameter finden Sie unter 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"
    }
  ]
}
```


Auflisten der Befehle eines Stacks (describe-commands)

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Verwenden Sie den Befehl describe-commands, um die Befehle eines Stacks aufzulisten oder detaillierte Informationen über die angegebenen Befehle zu erhalten. Im folgenden Beispiel werden Informationen über die auf einer bestimmten Instance ausgeführten Befehle ermittelt.

```
aws opsworks --region us-west-1 describe-commands --instance-id
 8c2673b9-3fe5-420d-9cfa-78d875ee7687
```

Der Befehl gibt ein JSON-Objekt mit Details über die einzelnen Befehle zurück. In diesem Beispiel identifiziert der Parameter Type den Befehl "Benennen", "Bereitstellen" oder "Bereitstellung aufheben". Eine Beschreibung der restlichen Parameter finden Sie unter 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
    }
  ]
}
```

Auflisten von Bereitstellungen eines Stacks (describe-deployments)

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Verwenden Sie den Befehl describe-deployments, um die Bereitstellungen eines Stacks aufzulisten oder detaillierte Informationen über die angegebenen Bereitstellungen zu erhalten.

```
aws opsworks --region us-west-1 describe-deployments --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

Der zuvor genannte Befehl gibt ein JSON-Objekt mit Details zu jeder Bereitstellung für den angegebenen Stack zurück. Eine Beschreibung aller Parameter finden Sie unter 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"
        }
    ]
}
```

Listet die Elastic IP-Adressen eines Stacks auf () describe-elastic-ips

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Verwenden Sie den describe-elastic-ipsBefehl, um die Elastic IP-Adressen aufzulisten, die bei einem Stack registriert wurden, oder um Details zu bestimmten Elastic IP-Adressen abzurufen.

```
aws opsworks --region us-west-2 describe-elastic-ips --instance-id b62f3e04-e9eb-436c-a91f-d9e9a396b7b0
```

Der zuvor genannte Befehl gibt ein JSON-Objekt mit Details zu den einzelnen Elastic IP-Adressen (in diesem Beispiel eine) für eine bestimmte Instance zurück. Eine Beschreibung der einzelnen Parameter finden Sie unter describe-elastic-ips.

```
{
  "ElasticIps": [
      {
          "Ip": "192.0.2.0",
          "Domain": "standard",
          "Region": "us-west-2"
      }
  ]
}
```

Auflisten der Instances eines Stacks (describe-Instances)

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Verwenden Sie den Befehl describe-instances, um die Instances eines Stacks aufzulisten oder detaillierte Informationen über die angegebenen Instances zu erhalten.

```
C:\>aws opsworks --region us-west-2 describe-instances --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

Der zuvor genannte Befehl gibt ein JSON-Objekt mit Informationen über alle Instances in einem bestimmten Stack zurück. Eine Beschreibung aller Parameter finden Sie unter 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"
        }
    ]
}
```

Auflisten der Stacks eines Kontos (describe-stacks)

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Verwenden Sie den Befehl describe-stacks, um die Stacks eines Kontos aufzulisten oder detaillierte Informationen über die angegebenen Stacks zu erhalten.

```
aws opsworks --region us-west-2 describe-stacks
```

Der zuvor genannte Befehl gibt ein JSON-Objekt mit Details zu jedem Stack des Kontos zurück, in diesem Beispiel zwei. Eine Beschreibung aller Parameter finden Sie unter 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"
        }
    ]
}
```

Auflisten der Layer eines Stacks (describe-layers)

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Verwenden Sie den Befehl describe-layers, um die Ebenen eines Stacks aufzulisten oder detaillierte Informationen über die angegebenen Ebenen zu erhalten.

```
aws opsworks --region us-west-2 describe-layers --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

Der vorherige Befehl gibt ein JSON-Objekt zurück, das Details zu jeder Ebene in einem angegebenen Stapel enthält — in diesem Beispiel eine MySQL-Ebene und eine benutzerdefinierte Ebene. Eine Beschreibung aller Parameter finden Sie unter 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": []
    }
  ]
}
```

Ausführen eines Rezepts (create-deployment)

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Verwenden Sie den Befehl create-deployment, um Stack-Befehle und Bereitstellungsbefehle auszuführen. Das folgende Beispiel führt einen Stack-Befehl zum Ausführen eines benutzerdefinierten Rezepts in einem bestimmten Stack aus.

```
aws opsworks --region us-west-1 create-deployment --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb
    --command "{\"Name\":\"execute_recipes\", \"Args\":{\"recipes\":[\"phpapp::appsetup
\"]}}"
```

Das Argument `command` greift auf ein folgendermaßen formatiertes JSON-Objekt zurück:

- `Name` - Gibt den Befehlsnamen an. Der in diesem Beispiel verwendete Befehl `execute_recipes` führt ein vorgegebenes Rezept auf den Instances des Stacks aus.
- `Args` - Gibt eine Liste der Argumente und deren Werte an. Dieses Beispiel enthält ein Argument, `recipes`, das gesetzt ist, um das Rezept auszuführen, `phpapp::appsetup`.

Beachten Sie, dass alle "-Zeichen im JSON-Objekt mit Escape-Zeichen versehen sind. Andernfalls gibt der Befehl möglicherweise eine Fehlermeldung aus, die besagt, dass JSON ungültig ist.

Der Befehl gibt eine Bereitstellungs-ID zurück, mit der Sie den Befehl für andere CLI-Befehle, wie beispielsweise `describe-commands`, identifizieren können.

```
{
    "DeploymentId": "5cbaa7b9-4e09-4e53-aa1b-314fbd106038"
}
```

Installieren von Abhängigkeiten (create-deployment)

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Verwenden Sie den Befehl create-deployment, um Stack-Befehle und Bereitstellungsbefehle auszuführen. Im folgenden Beispiel wird der Stack-Befehl `update_dependencies` ausgeführt, um die Abhängigkeiten auf den Instances eines Stacks zu aktualisieren.

```
aws opsworks --region us-west-1 create-deployment --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb
--command "{\"Name\":\"install_dependencies\"}"
```

Das Argument `command` greift auf ein JSON-Objekt mit einem Parameter `Name` zurück, dessen Wert den Befehlsnamen, in diesem Beispiel `install_dependencies`, angibt. Beachten Sie, dass alle "-Zeichen im JSON-Objekt mit Escape-Zeichen versehen sind. Andernfalls gibt der Befehl möglicherweise eine Fehlermeldung aus, die besagt, dass JSON ungültig ist.

Der Befehl gibt eine Bereitstellungs-ID zurück, mit der Sie den Befehl für andere CLI-Befehle, wie beispielsweise `describe-commands`, identifizieren können.

```
{
    "DeploymentId": "aef5b255-8604-4928-81b3-9b0187f962ff"
}
```

Aktualisieren der Stack-Konfiguration (update-stack)

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Verwenden Sie den Befehl update-stack, um die Konfiguration eines bestimmten Stacks zu aktualisieren. Das folgende Beispiel aktualisiert einen Stack, um ein benutzerdefiniertes JSON-Objekt zu den Stack-Konfigurationsattributen hinzuzufügen.

```
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
```

Beachten Sie, dass alle "-Zeichen im JSON-Objekt mit Escape-Zeichen versehen sind. Andernfalls gibt der Befehl möglicherweise eine Fehlermeldung aus, die besagt, dass JSON ungültig ist.

Note

Das Beispiel gibt auch eine Service-Rolle für den Stack an. Sie müssen `service-role-arn` auf einen gültige ARN-Service setzen oder die Aktion wird fehlschlagen, da kein Standardwert existiert. Sie können die aktuelle ARN-Service-Rolle angeben, wenn Sie das bevorzugen, allerdings müssen Sie das dann explizit tun.

Der Befehl `update-stack` gibt keinen Wert zurück.

Handbuch zur Fehlersuche und Fehlerbehebung

Important

Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn Sie Probleme bei einem Rezept oder einem Service beheben müssen, arbeiten Sie nacheinander die folgenden Punkte ab:

1. Überprüfen Sie für Ihr Problem Debugging und Fehlerbehebung bei bekannten Problemen.
2. Durchsuchen Sie das AWS OpsWorks Stacks-Forum nach Antworten auf Ihre Frage.

   Das Forum umfasst viele erfahrene Benutzer und wird vom AWS OpsWorks Stacks-Team überwacht.
3. Informationen zu Problemen mit Rezepten finden Sie unter Debuggen von Rezepten.
4. Wenden Sie sich an den AWS OpsWorks Stacks-Support oder posten Sie Ihr Problem im AWS OpsWorks Stacks-Forum.

Der folgende Abschnitt enthält Hilfestellung zur Fehlerbehebung bei Rezepten. Im letzten Abschnitt werden häufige Probleme und deren Lösungen vorgestellt.

> **Note**
>
> Bei jeder Chef-Ausführung wird eine Protokolldatei erstellt, die eine detaillierte Beschreibung der Ausführung enthält und eine wertvolle Informationsquelle zur Fehlerbehebung ist. Um den Detailgrad des Protokolls festzulegen, fügen Sie eine `Chef::Log.level`-Anweisung zu einem benutzerdefinierten Rezept hinzu, um die Protokollebene festzulegen. Der Standardwert ist `:info`. Im folgenden Beispiel sehen Sie, wie Sie die Chef-Protokollebene auf `:debug` festlegen, um möglichst detaillierte Informationen zu erhalten.
>
> ```
> Chef::Log.level = :debug
> ```
>
> Weitere Informationen zum Anzeigen und Deuten von Chef-Protokollen finden Sie unter Chef-Protokolle.

Themen

- Debuggen von Rezepten
- Debugging und Fehlerbehebung bei bekannten Problemen

Debuggen von Rezepten

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn ein Lebenszyklusereignis auftritt oder Sie den Stack-Befehl "Rezepte ausführen" ausführen, gibt AWS OpsWorks Stacks einen Befehl an den Agenten aus, um eine Chef Solo-Ausführung auf den angegebenen Instances zu starten und das entsprechende Rezept, einschließlich Ihrer benutzerdefinierten Rezepte, auszuführen. Dieser Abschnitt beschreibt einige Möglichkeiten zum Debuggen fehlgeschlagener Rezepte.

Themen

- Anmelden bei einer fehlgeschlagenen Instance
- Chef-Protokolle
- Verwenden der AWS OpsWorks Stacks Agent CLI

Anmelden bei einer fehlgeschlagenen Instance

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Wenn ein Rezept fehlschlägt, nimmt die Instance den Zustand `setup_failed` anstatt "online" an. Auch wenn die Instance für AWS OpsWorks Stacks nicht online ist, läuft die EC2-Instance und es ist oft nützlich, sich anzumelden, um das Problem zu beheben. Beispielsweise können Sie prüfen,

ob eine Anwendung oder ein benutzerdefiniertes Rezeptbuch korrekt installiert ist. Die in AWS OpsWorks Stacks integrierte Unterstützung für SSH - und RDP-Anmeldung ist nur für Instances im Online-Status verfügbar. Wenn Sie jedoch ein SSH-Schlüsselpaar der Instance zugewiesen haben, können Sie sich trotzdem wie folgt anmelden:

- Linux-Instances — Verwenden Sie den privaten Schlüssel des SSH-Schlüsselpaars, um sich mit einem SSH-Client eines Drittanbieters wie OpenSSH oder PuTTY anzumelden.

  Sie können für diesen Zweck ein EC2-Schlüsselpaar oder Ihr persönliches SSH-Schlüsselpaar verwenden.
- Windows-Instances — Verwenden Sie den privaten Schlüssel des EC2-Schlüsselpaars, um das Administratorkennwort der Instance abzurufen.

  Verwenden Sie dieses Passwort, um sich mit Ihrem bevorzugten RDP-Client anzumelden. Weitere Informationen finden Sie unter Anmelden als Administrator. Sie können kein persönliches SSH-Schlüsselpaar nutzen, um ein Administratorpasswort abzurufen.

Chef-Protokolle

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Chef-Logs sind eine Ihrer wichtigsten Ressourcen zur Fehlerbehebung, insbesondere beim Debuggen von Rezepten. AWS OpsWorks Stacks erfasst das Chef-Protokoll für jeden Befehl und speichert die Protokolle für die 30 neuesten Befehle einer Instanz. Da sich die Ausführung im Debug-Modus befindet, enthält das Protokoll eine detaillierte Beschreibung der Chef-Ausführung, einschließlich des Textes, der nach `stdout` und `stderror` gesendet wird. Wenn ein Rezept fehlschlägt, enthält das Protokoll den Chef-Stack-Trace.

AWS OpsWorks Stacks bietet Ihnen mehrere Möglichkeiten, Chef-Logs einzusehen. Nachdem Sie über die Protokollinformationen verfügen, können Sie sie verwenden, um fehlgeschlagene Rezepte zu debuggen.

> **Note**
>
> Sie können sich auch ein bestimmtes Protokollfragment anzeigen lassen, indem Sie SSH verwenden, um eine Verbindung mit der Instance herzustellen und den Agenten-CLI-Befehl `show_log` auszuführen. Weitere Informationen finden Sie unter Anzeigen von Chef-Protokollen.

Themen

- Anzeige eines Chef-Protokolls mit der Konsole
- Anzeige eines Chef-Protokolls mit CLI oder API
- Anzeige eines Chef-Protokolls auf einer Instance
- Interpretieren eines Chef-Protokolls
- Häufige Chef-Protokollfehler

Anzeige eines Chef-Protokolls mit der Konsole

Die einfachste Möglichkeit, ein Chef-Protokoll anzuzeigen, ist auf die Instance-Detailseite zu gehen. Der Abschnitt Logs (Protokolle) enthält einen Eintrag für jedes Ereignis und den Befehl Execute Recipes (Rezepte ausführen). Das folgende Beispiel zeigt den Abschnitt Logs (Protokolle) einer Instance mit den Befehlen configure (Konfigurieren) und setup (Einrichten), die dem Konfigurieren und Einrichten von Lebenszyklusereignissen entsprechen.

Klicken Sie auf den entsprechenden Befehl show (Anzeigen) in der Spalte Log (Protokoll), um das entsprechende Chef-Protokoll anzuzeigen. Wenn ein Fehler auftritt, öffnet AWS OpsWorks Stacks automatisch das Protokoll mit dem Fehler, das sich normalerweise am Ende der Datei befindet.

Anzeige eines Chef-Protokolls mit CLI oder API

Sie können den AWS OpsWorks `describe-commands`Stacks-CLI-Befehl oder die `DescribeCommands`API-Aktion verwenden, um die Protokolle anzuzeigen, die in einem Amazon S3

S3-Bucket gespeichert sind. Das folgende Beispiel zeigt, wie Sie mit der Befehlszeilenschnittstelle (CLI) jeden der aktuellen Protokolldateisätze für eine bestimmte Instance anzeigen. Das Verfahren für die Nutzung von `DescribeCommands` ist im Wesentlichen ähnlich.

Um die AWS OpsWorks Stacks zum Anzeigen der Chef-Logs einer Instance zu verwenden

1. Öffnen Sie die Detailseite der Instanz und kopieren Sie ihren OpsWorksID-Wert.
2. Verwenden Sie den ID-Wert, um den CLI-Befehl `describe-commands` wie folgt auszuführen:

```
aws opsworks describe-commands --instance-id 67bf0da2-29ed-4217-990c-d895d51812b9
```

Der Befehl gibt für jeden Befehl, den AWS OpsWorks Stacks auf der Instanz ausgeführt hat, ein JSON-Objekt mit einem eingebetteten Objekt zurück, wobei der neueste Befehl an erster Stelle steht. Der Parameter `Type` enthält den Befehlstyp für jedes eingebettete Objekt, in diesem Beispiel einen Befehl `configure` und einen Befehl `setup`.

```
{
    "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. Kopieren Sie den Wert `LogUrl` in den Browser, um das Protokoll anzuzeigen.

Wenn die Instance mehr als einige Befehle enthält, können Sie `describe-commands` Parameter hinzufügen, um zu filtern, welche Befehle in dem Antwortobjekt enthalten sind. Weitere Informationen finden Sie unter describe-commands.

Anzeige eines Chef-Protokolls auf einer Instance

> **Note**
>
> Die Themen in diesem Abschnitt gelten für Chef 12. Weitere Informationen zu dem Speicherort von Chef-Protokollen für Chef 11.10 und niedriger finden Sie unter Troubleshooting Chef 11.10 and Earlier Versions for Linux (Fehlerbehebung bei Chef 11.10 und früheren Versionen für Linux).

Linux-Instances

AWS OpsWorks Stacks speichert die Chef-Logs jeder Instanz in ihrem `/var/chef/runs` Verzeichnis. (Dieses Verzeichnis umfasst für Linux-Instances auch die zugehörigen Datenbehälter, die als JSON-formatierte Dateien gespeichert werden.) Sie benötigen Sudo-Berechtigungen, um auf dieses Verzeichnis zuzugreifen. Das Protokoll für jede Ausführung befindet sich in einer Datei mit dem Namen `chef.log` innerhalb des Unterverzeichnisses, das einzeln ausgeführt wird.

AWS OpsWorks Stacks speichert seine internen Protokolle im Ordner der `/var/log/aws/opsworks` Instanz. Die Informationen sind in der Regel nicht sehr hilfreich für Fehlerbehebungszwecke. Diese Protokolle sind jedoch für den AWS OpsWorks Stacks-Support nützlich, und Sie werden möglicherweise aufgefordert, sie bereitzustellen, wenn Sie auf ein

Problem mit dem Dienst stoßen. Die Linux-Protokolle können manchmal auch nützliche Daten zur Fehlerbehebung liefern.

Windows-Instances

Agenten-Protokolle

Auf Windows-Instanzen werden OpsWorks Protokolle in einem `ProgramData` Pfad wie dem folgenden gespeichert. Die Anzahl umfasst einen Zeitstempel.

```
C:\ProgramData\OpsWorksAgent\var\logs\number
```

> **Note**
>
> Standardmäßig handelt es sich bei `ProgramData` um einen versteckten Ordner. Navigieren Sie zu Folder Options (Ordneroptionen), um ihn wieder einzublenden. Klicken Sie unter View (Anzeigen) auf die Option zum Anzeigen der ausgeblendeten Dateien.

Das folgende Beispiel zeigt Agenten-Protokolle auf einer Windows-Instance.

```
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
```

Chef-Protokolle

OpsWorks-Protokolle werden auf Windows-Instances unter dem Pfad `ProgramData` wie folgt gespeichert. Die Anzahl umfasst einen Zeitstempel.

```
C:\ProgramData\OpsWorksAgent\var\commands\number
```

> **Note**
>
> Dieses Verzeichnis enthält nur die Ausgabe des ersten (OpsWorks eigenen) Chef-Laufs.

Das folgende Beispiel zeigt OpsWorks eigene Chef-Protokolle auf einer Windows-Instanz.

```
 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
```

Benutzer-Chef-Protokolle

Die Protokolle für Ihre Chef-Ausführungen finden Sie in den Dateien namens `logfile.txt` in einem Ordner, der nach dem nummerierten Chef-Befehl, wie im folgenden Diagramm, benannt wird.

C:/chef └── runs └── command-12345 ├── attribs.json ├── client.rb └── logfile.txt

Interpretieren eines Chef-Protokolls

Der Anfang des Protokolls besteht hauptsächlich aus einer internen Chef-Protokollierung.

```
# 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
...
```

Dieser Teil der Datei ist vor allem für Chef-Experten nützlich. Beachten Sie, dass die Ausführungsliste nur zwei Rezepte enthält, obwohl die meisten Befehle viele mehr umfassen. Diese beiden Rezepte bewältigen die Aufgabe, das Laden und die Ausführung aller anderen integrierten und benutzerdefinierten Rezepte durchzuführen.

Der interessanteste Teil der Datei befindet sich hauptsächlich am Ende. Wenn eine Ausführung erfolgreich endet, sollten Sie etwas wie das Folgende sehen:

```
...
[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**
>
> Sie können die Agenten-CLI verwenden, um das Protokollfragment während oder nach der Ausführung anzuzeigen. Weitere Informationen finden Sie unter Anzeigen von Chef-Protokollen.

Wenn ein Rezept fehlschlägt, sollten Sie nach einer Ausgabe auf ERROR-Ebene suchen, die eine Ausnahme gefolgt von einem Chef-Stack-Trace enthalten wird, wie z. B. die folgende:

```
...
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
```

Das Dateiende ist der Chef-Stack-Trace. Sie sollten auch die Ausgabe direkt vor der Ausnahme überprüfen, da sie oft einen Systemfehler, wie z. B. `package not available` enthält, der auch nützlich sein kann, die Ursache des Fehlers zu ermitteln. In diesem Fall konnte MySQL-Daemon nicht gestartet werden.

Häufige Chef-Protokollfehler

Es folgen einige gängige Chef-Protokollfehler und wie diese behoben werden.

Das Protokoll konnte nicht gefunden werden

Zu Beginn eines Chef-Laufs erhalten Instances eine vorsignierte Amazon S3 S3-URL, mit der Sie das Protokoll auf einer Webseite anzeigen können, wenn der Chef-Lauf abgeschlossen ist. Da diese URL nach zwei Stunden abläuft, wird kein Protokoll auf die Amazon S3 S3-Website hochgeladen, wenn ein Chef-Lauf länger als zwei Stunden dauert, auch wenn während des Chef-Laufs keine Probleme aufgetreten sind. Der Befehl zum Erstellen eines Protokolls wird erfolgreich ausgeführt, aber das Protokoll kann nur auf der Instance angezeigt werden und nicht auf der vorsignierten URL.

Das Protokoll endet abrupt

Wenn ein Chef-Protokoll abrupt endet, ohne entweder einen Erfolg anzugeben oder eine Fehlerinformation anzuzeigen, ist wahrscheinlich ein niedriger Speicherstatus aufgetreten, der Chef daran hindert, das Protokoll abzuschließen. Die einfachste Lösung ist der erneute Versuch mit einer größeren Instance.

Fehlendes Rezeptbuch oder Rezept

Wenn die Chef-Ausführung ein Rezeptbuch oder ein Rezept antrifft, das nicht im Rezeptbuchzwischenspeicher vorhanden ist, sehen Sie Folgendes:

```
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?
```

Dieser Eintrag gibt an, dass sich das Rezeptbuch `mycookbook` nicht im Rezeptbuchzwischenspeicher befindet. Mit Chef 11.4 kann dieser Fehler auch auftreten, wenn Sie Abhängigkeiten in `metadata.rb` nicht korrekt angeben.

AWS OpsWorks Stacks führt Rezepte aus dem Kochbuch-Cache der Instanz aus. Es lädt Rezeptbücher von Ihrem Repository auf diesen Zwischenspeicher herunter, wenn die Instance gestartet wird. AWS OpsWorks Stacks aktualisiert den Cache einer Online-Instanz jedoch nicht automatisch, wenn Sie die Kochbücher in Ihrem Repository nachträglich ändern. Wenn Sie seit dem Start der Instance Ihre Rezeptbücher geändert oder neue Rezeptbücher hinzugefügt haben, führen Sie die folgenden Schritte aus:

1. Stellen Sie sicher, dass Sie Ihre Änderungen an das Repository übergeben haben.
2. Führen Sie den Stack-Befehl Update Cookbooks (Rezeptbücher aktualisieren) aus, um den Rezeptbuchzwischenspeicher mit der jeweils aktuellen Version aus dem Repository zu aktualisieren.

Lokaler Befehlsausfall

Wenn bei einer Chef-Ressource `execute` die Ausführung des angegebenen Befehls fehlschlägt, sehen Sie Folgendes:

```
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=/
```

Navigieren Sie in dem Protokoll nach oben. Sie sollten die Ausgaben `stderr` und `stdout` des Befehls sehen, mit denen Sie ermitteln können, warum der Befehl fehlgeschlagen ist.

Paketausfall

Wenn eine Paketinstallation fehlschlägt, sehen Sie Folgendes:

```
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
```

Navigieren Sie in dem Protokoll nach oben. Sie sollten eine STDOUT- und STDERROR-Ausgabe des Befehls sehen, mit denen Sie ermitteln können, warum die Paketinstallation fehlgeschlagen ist.

Verwenden der AWS OpsWorks Stacks Agent CLI

Important

Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Note

Die Agenten-CLI ist nur für Linux-Instances verfügbar.

Auf jeder Online-Instanz installiert AWS OpsWorks Stacks einen Agenten, der mit dem Dienst kommuniziert. Der AWS OpsWorks Stacks-Dienst wiederum sendet Befehle an den Agenten,

um Aufgaben wie das Initiieren von Chef-Läufen auf der Instanz auszuführen, wenn ein Lebenszyklusereignis eintritt. Auf Linux-Instances stellt der Agent eine Befehlszeilenschnittstelle (Command Line Interface, CLI) bereit, die für die Fehlerbehebung sehr nützlich ist. Um Agenten-CLI-Befehle auszuführen, verwenden Sie SSH , um eine Verbindung mit einer Instance herzustellen. Anschließend können Sie Agenten-CLI-Befehle ausführen, um eine Vielzahl an Aufgaben durchzuführen, einschließlich der folgenden:

- Rezepte ausführen.
- Chef-Protokolle anzeigen.
- -Stack-Konfiguration und JSON-Bereitstellung anzeigen.

Weitere Informationen zum Einrichten einer SSH-Verbindung zu einer Instance finden Sie unter Anmelden mit SSH. Zudem müssen Sie über SSH- und Sudo-Berechtigungen für den Stack verfügen.

In diesem Abschnitt wird beschrieben, wie Sie die Agenten-CLI für die Fehlerbehebung verwenden. Weitere Informationen und eine vollständige Befehlsreferenz finden Sie unter AWS OpsWorks Stacks Agent CLI.

Themen

- Ausführen von Rezepten
- Anzeigen von Chef-Protokollen
- Anzeigen der Stack-Konfiguration und der JSON-Bereitstellung

Ausführen von Rezepten

Der Agenten-CLI-Befehl `run_command` weist den Agenten an, einen bereits zuvor durchgeführten Befehl erneut auszuführen. Die wichtigsten Befehle für die Fehlerbehebung – `setup`, `configure`, `deploy` und `undeploy` – entsprechen jeweils einem Lebenszyklusereignis. Sie weisen den Agenten an, eine Chef-Ausführung zu initiieren, um die zugehörigen Rezepte auszuführen.

> **Note**
>
> Der Befehl `run_command` ist auf die Ausführung der Rezeptgruppe begrenzt, die einem bestimmten Befehl zugeordnet ist, in der Regel die Rezepte, die mit einem Lebenszyklusereignis verknüpft sind. Sie können ihn nicht verwenden, um ein bestimmtes

Rezept auszuführen. Um ein oder mehrere angegebene Rezepte auszuführen, verwenden Sie den Stack-Befehl Execute Recipes (Rezepte ausführen) oder die entsprechenden CLI- oder API-Aktionen (`create-deployment` und `CreateDeployment`).

Der Befehl `run_command` ist für das Debuggen benutzerdefinierter Rezepte nützlich, besonders Rezepten, die der Einrichtung und Konfiguration von Lebenszyklusereignissen zugewiesen sind, die Sie nicht direkt über die Konsole auslösen können. Durch die Verwendung von `run_command` können Sie die Rezepte eines bestimmten Ereignisses so häufig wie nötig ausführen, ohne dass Sie Instances starten oder beenden müssen.

Note

AWS OpsWorks Stacks führt Rezepte aus dem Kochbuch-Cache der Instanz aus, nicht aus dem Kochbuch-Repository. AWS OpsWorks Stacks lädt beim Start der Instanz Kochbücher in diesen Cache herunter, aktualisiert den Cache auf Online-Instanzen jedoch nicht automatisch, wenn Sie Ihre Kochbücher nachträglich ändern. Wenn Sie Ihre Rezeptbücher seit dem Start der Instance geändert haben, müssen Sie den Stack-Befehl zum Aktualisieren der Rezeptbücher ausführen, um den Rezeptbuchzwischenspeicher mit der neuesten Version aus dem Repository zu aktualisieren.

Nur die neuesten Befehle werden vom Agenten zwischengespeichert. Sie können sie auflisten, indem Sie `list_commands` ausführen, welches eine Liste der zwischengespeicherten Befehle und die Zeit, in der die Operationen ausgeführt wurden, zurückgibt.

```
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
```

Um den neuesten Befehl erneut auszuführen, führen Sie dies aus:

```
sudo opsworks-agent-cli run_command
```

Um die neueste Instance eines angegebenen Befehls erneut auszuführen, führen Sie dies aus:

```
sudo opsworks-agent-cli run_command command
```

Um beispielsweise die Schritte zum Einrichten der Rezepte erneut durchzuführen, können Sie den folgenden Befehl ausführen:

```
sudo opsworks-agent-cli run_command setup
```

Jeder Befehl verfügt über eine zugewiesene Stack-Konfiguration und JSON-Bereitstellung, in der der Stack- und Bereitstellungsstatus zum Zeitpunkt der Befehlsausführung angegeben ist. Da sich diese Daten von einem Befehl zum nächsten ändern können, kann eine ältere Instance eines Befehls etwas andere Daten verwenden als die neueste. Um eine bestimmte Instance eines Befehls erneut auszuführen, kopieren Sie die Zeit von der Ausgabe `list_commands` und führen Sie die folgenden Schritte aus:

```
sudo opsworks-agent-cli run_command time
```

Die vorherigen Beispiele führen alle den Befehl erneut aus, indem sie das Standard-JSON-Format verwenden, das heißt, dass das JSON-Format für diesen Befehl installiert wurde. Sie können einen Befehl anhand einer beliebigen JSON-Datei wie folgt erneut ausführen:

```
sudo opsworks-agent-cli run_command -f /path/to/valid/json.file
```

Anzeigen von Chef-Protokollen

Der Agenten-CLI-Befehl `show_log` zeigt ein bestimmtes Protokoll an. Nachdem der Befehl abgeschlossen ist, werden Sie das Dateiende sehen. Der Befehl `show_log` bietet daher eine bequeme Möglichkeit, das Protokoll zu fragmentieren, in dem Sie normalerweise die Fehlerinformationen finden. Sie können nach oben navigieren, um die früheren Teile des Protokolls zu sehen.

Um das aktuelle Protokoll des Befehls einzusehen, führen Sie dies aus:

```
sudo opsworks-agent-cli show_log
```

Sie können auch Protokolle für einen bestimmten Befehl anzeigen, beachten Sie jedoch, dass der Agent nur Protokolle der letzten 30 Befehle zwischenspeichert. Sie können Befehle einer Instance auflisten, indem Sie `list_commands` ausführen, wodurch eine Liste der zwischengespeicherten

Befehle und die Zeit, in der die Operationen ausgeführt wurden, zurückgegeben werden. Ein Beispiel finden Sie unter Ausführen von Rezepten.

Um das Protokoll für die neueste Ausführung eines bestimmten Befehls anzuzeigen, führen Sie die folgenden Schritte aus:

```
sudo opsworks-agent-cli show_log command
```

Der Befehlsparameter kann auf `setup`, `configure`, `deploy`, `undeploy`, `start`, `stop` oder `restart` gesetzt werden. Die meisten dieser Befehle entsprechen Lebenszyklusereignissen und weisen den Agenten an, die zugehörigen Rezepte auszuführen.

Um das Protokoll für eine bestimmte Befehlsausführung anzuzeigen, kopieren Sie das Datum aus der Ausgabe `list_commands` und führen Folgendes aus:

```
sudo opsworks-agent-cli show_log date
```

Wenn ein Befehl noch ausgeführt wird, zeigt `show_log` den aktuellen Zustand des Protokolls an.

> **Note**
>
> Eine Möglichkeit, Fehler und out-of-memory Probleme `show_log` zu beheben, besteht darin, ein Protokoll während der Ausführung wie folgt zu protokollieren:
>
> 1. Verwenden Sie `run_command`, um das entsprechende Lebenszyklusereignis auszulösen. Weitere Informationen finden Sie unter Ausführen von Rezepten.
> 2. Führen Sie `show_log` mehrmals aus, um zu sehen, wie das Protokollfragment geschrieben wird.
>
> Wenn Chef nicht genügend Arbeitsspeicher zur Verfügung steht oder es unerwartet beendet wird, wird das Protokoll abrupt beendet. Wenn ein Rezept fehlschlägt, wird das Protokoll mit einer Ausnahme und einem Stack-Trace beendet.

Anzeigen der Stack-Konfiguration und der JSON-Bereitstellung

Ein Großteil der Daten, die von Rezepten verwendet werden, stammen von der Stack-Konfiguration und JSON-Bereitstellung, die eine Reihe von Chef-Attributen definieren, die eine detaillierte

Beschreibung der Stack-Konfiguration, alle Bereitstellungen und optionale benutzerdefinierte Attribute, die Benutzer hinzufügen können, bereitstellen. Für jeden Befehl installiert AWS OpsWorks Stacks eine JSON-Datei, die den Stack und den Bereitstellungsstatus zum Zeitpunkt des Befehls darstellt. Weitere Informationen finden Sie unter Attribute für die Stack-Konfiguration und -Bereitstellung.

Wenn Ihre benutzerdefinierten Rezepte die Daten aus der Stack-Konfiguration und JSON-Bereitstellung erhalten, können Sie die Daten überprüfen, indem Sie das JSON-Objekt überprüfen. Die einfachste Methode zur Anzeige der Stack-Konfiguration und JSON-Bereitstellung ist, den Agenten-CLI-Befehl `get_json` auszuführen, der eine formatierte Version des JSON-Objekts anzeigt. Das folgende Beispiel zeigt die ersten Zeilen einiger typischer Ausgaben:

```
{
  "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"
...
```

Sie können die neueste Stack-Konfiguration und JSON-Bereitstellung wie folgt anzeigen:

```
sudo opsworks-agent-cli get_json
```

Sie können die neueste Stack-Konfiguration und JSON-Bereitstellung für einen bestimmten Befehl anzeigen, indem Sie den folgenden Befehl ausführen:

```
sudo opsworks-agent-cli get_json command
```

Der Befehlsparameter kann auf `setup`, `configure`, `deploy`, `undeploy`, `start`, `stop` oder `restart` gesetzt werden. Die meisten dieser Befehle entsprechen Lebenszyklusereignissen und weisen den Agenten an, die zugehörigen Rezepte auszuführen.

Sie können die Stack-Konfiguration und JSON-Bereitstellung für eine bestimmte Befehlsausführung anzeigen, indem Sie das Datum des Befehls wie folgt angeben:

```
sudo opsworks-agent-cli get_json date
```

Die einfachste Möglichkeit zur Verwendung dieses Befehls ist:

1. Führen Sie `list_commands` aus, welches eine Liste der Befehle, die in der Instance ausgeführt wurden, und das Datum, an dem jeder Befehl ausgeführt wurde, zurückgibt.
2. Kopieren Sie das Datum für den entsprechenden Befehl und nutzen Sie es als das `get_json` *`Datums`*-Argument.

Debugging und Fehlerbehebung bei bekannten Problemen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

In diesem Abschnitt werden das Debugging und die Fehlerbehebung bei bekannten Problemen beschrieben.

Themen

- Fehlerbehebung bei Stacks AWS OpsWorks
- Fehlerbehebung bei der Instance-Registrierung

Fehlerbehebung bei Stacks AWS OpsWorks

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Dieser Abschnitt enthält einige häufig auftretende AWS OpsWorks Stacks-Probleme und deren Lösungen.

Themen

Instances können nicht verwaltet werden

Problem: Eine Instance, die bisher verwaltbar war, lässt sich nicht mehr verwalten. In einigen Fällen können Protokolle eine Fehlermeldung wie die folgende anzeigen.

```
Aws::CharlieInstanceService::Errors::UnrecognizedClientException - The security token
 included in the request is invalid.
```

Ursache: Dies kann auftreten, wenn eine Ressource, von der die Instance abhängig ist, außerhalb von AWS OpsWorks bearbeitet oder gelöscht wurde. Nachfolgend finden Sie Beispiele für Ressourcenänderungen, die zu einem Kommunikationsabbruch mit der Instance führen.

- Ein der Instance zugeordneter IAM-Benutzer oder eine IAM-Rolle wurde versehentlich außerhalb von AWS OpsWorks Stacks gelöscht. Dies führt zu einem Kommunikationsfehler zwischen dem AWS OpsWorks Agenten, der auf der Instance installiert ist, und dem AWS OpsWorks Stacks-Dienst. Der -Benutzer, der einer Instance zugeordnet ist, muss während der gesamten Instance-Lebensdauer vorhanden sein.
- Das Bearbeiten von Volume- oder Speicherkonfigurationen, während eine Instance offline ist, kann dazu führen, dass die Instance nicht mehr verwaltbar ist.
- Manuelles Hinzufügen von EC2-Instances zu einem ELB. AWS OpsWorks konfiguriert einen zugewiesenen Elastic Load Balancing Load Balancer jedes Mal neu, wenn eine Instance den Online-Status erreicht oder verlässt. AWS OpsWorks betrachtet nur Instances, von denen sie weiß, als gültige Mitglieder. Instances AWS OpsWorks, die außerhalb oder durch einen anderen Prozess hinzugefügt wurden, werden entfernt. Jede andere Instance wird ebenfalls entfernt.

Lösung: Löschen Sie keine IAM-Benutzer oder -Rollen, von denen Ihre Instanzen abhängen. Bearbeiten Sie (wenn möglich) die Volume- oder Speicherkonfigurationen nur, während die abhängigen Instances ausgeführt werden. Wird verwendet AWS OpsWorks , um die Load Balancer- oder EIP-Mitgliedschaften von Instances zu verwalten. AWS OpsWorks Um Probleme bei der Verwaltung registrierter Instances zu vermeiden, falls der Benutzer versehentlich gelöscht wird, fügen Sie bei der Registrierung einer Instance den `--use-instance-profile` Parameter zu Ihrem `register` Befehl hinzu, um stattdessen das integrierte Instanzprofil der Instanz zu verwenden.

Nach einer Chef-Ausführung starten die Instances nicht

Problem: Wenn in Chef 11.10 (oder einer älteren Version) Stacks für die Verwendung von benutzerdefinierten Rezeptbüchern konfiguriert sind, starten die Instances nach einer Chef-Ausführung mit Community-Rezeptbüchern nicht mehr. Die Protokollnachrichten besagen, dass die Rezepte nicht kompiliert ("Recipe Compile Error") oder geladen werden können (weil eine Abhängigkeit nicht gefunden wird).

Ursache: Vermutlich wird die vom Stack verwendete Chef-Version von den benutzerdefinierten oder Community-Rezeptbüchern nicht unterstützt. Bei einigen beliebten Community-Rezeptbüchern wie `apt` und `build-essential` sind Kompatibilitätsprobleme mit Chef 11.10 bekannt.

Lösung: Bei AWS OpsWorks Stacks-Stacks, bei denen die Einstellung Benutzerdefinierte Chef-Kochbücher verwenden aktiviert ist, müssen benutzerdefinierte Kochbücher oder Community-Kochbücher immer die Version von Chef unterstützen, die Ihr Stack verwendet. Legen Sie für die Community-Rezeptbücher eine Version fest (das heißt, Sie legen die Rezeptbuch-Versionsnummer auf eine bestimmte Version fest), die mit der Chef-Version kompatibel ist, die Sie in Ihren Stack-Einstellungen konfiguriert haben. Um eine unterstützte Version für Community-Rezeptbücher zu ermitteln, suchen Sie im Änderungsprotokoll nach einem Rezeptbuch, das nicht kompiliert werden konnte, und verwenden Sie nur die neueste Version des Rezeptbuchs, die von Ihrem Stack unterstützt wird. Um eine Rezeptbuchversion festzulegen, geben Sie im Repository des benutzerdefinierten Rezeptbuchs in der Berksfile-Datei die exakte Versionsnummer an. z. B. `cookbook 'build-essential', '= 3.2.0'`.

Die Instances eines Layers bestehen alle bei einem Elastic Load Balancing Health Check nicht

Problem: Sie fügen einen Elastic Load Balancing Load Balancer an eine App-Serverebene an, aber alle Instances bestehen die Integritätsprüfung nicht.

Ursache: Wenn Sie einen Elastic Load Balancing Load Balancer erstellen, müssen Sie den Ping-Pfad angeben, den der Load Balancer aufruft, um festzustellen, ob die Instance fehlerfrei ist. Stellen Sie sicher, dass der Ping-Pfad für die Anwendung geeignet ist. Der Standardwert lautet "/index.html". Falls die Anwendung den Wert `index.html` nicht enthält, müssen Sie einen entsprechenden Pfad angeben. Andernfalls schlägt die Zustandsprüfung fehl. Beispielsweise verwendet die im Thema Erste Schritte mit Chef 11 Linux-Stacks verwendete SimplePHPApp-Anwendung nicht den Wert "index.html". Der geeignete Ping-Pfad für diese Server lautet "/".

Lösung: Ändern Sie den Ping-Pfad für den Load Balancer. Weitere Informationen finden Sie unter Elastic Load Balancing

Es kann nicht mit einem Elastic Load Balancing Load Balancer kommuniziert werden

Problem: Sie erstellen einen Elastic Load Balancing Load Balancer und hängen ihn an eine App-Serverebene an. Wenn Sie jedoch auf den DNS-Namen oder die IP-Adresse des Load Balancers klicken, um die Anwendung auszuführen, erhalten Sie die folgende Fehlermeldung: „Der Remote-Server reagiert nicht“.

Ursache: Wenn Ihr Stack in einer Standard-VPC ausgeführt wird, müssen Sie beim Erstellen eines Elastic Load Balancing-Load Balancers in der Region eine Sicherheitsgruppe angeben. Die für den Dateneingang festgelegten Regeln der Sicherheitsgruppe müssen eingehenden Datenverkehr von Ihrer IP-Adresse zulassen. Wenn Sie default VPC security group (standardmäßige VPC-Sicherheitsgruppe) angeben, akzeptiert die Standardregel keinen eingehenden Datenverkehr.

Lösung: Ändern Sie die Dateneingangsregeln für die Sicherheitsgruppe, damit eingehender Datenverkehr von den entsprechenden IP-Adressen akzeptiert wird.

1. Klicken Sie im Navigationsbereich der Amazon EC2 EC2-Konsole auf Sicherheitsgruppen.
2. Wählen Sie die Sicherheitsgruppe für den Load Balancer aus.
3. Klicken Sie auf Edit (Bearbeiten) auf der Registerkarte Inbound (Eingehend).
4. Fügen Sie eine Dateneingangsregel hinzu und legen Sie Source (Quelle) auf einen geeigneten CIDR-Wert fest.

   Wenn Sie beispielsweise Anywhere (Beliebig) angeben, wird der CIDR-Wert auf 0.0.0.0/0 festgelegt, sodass der Load Balancer eingehenden Datenverkehr von einer beliebigen IP-Adresse akzeptiert.

Für eine importierte lokale Instance kann die Volume-Einrichtung nach einem Neustart nicht abgeschlossen werden

Problem: Sie starten eine EC2-Instance neu, die Sie in AWS OpsWorks Stacks importiert haben, und in der AWS OpsWorks Stacks-Konsole wird als Instance-Status „Failed“ angezeigt. Dies kann bei Chef 11- oder Chef 12-Instances auftreten.

Ursache: AWS OpsWorks Stacks kann vermutlich im Rahmen der Einrichtung kein Volume für die Instance hinzufügen. Ein möglicher Grund ist, dass AWS OpsWorks Stacks bei Ausführung des Befehls `setup` die Volume-Konfiguration der Instance überschreibt.

Lösung: Öffnen Sie die Seite Details für die Instance und überprüfen Sie die Volume-Konfiguration im Bereich Volumes. Beachten Sie, dass Sie die Volume-Konfiguration nur ändern können, wenn die Instance den Status stopped (angehalten) aufweist. Stellen Sie sicher, dass jedes Volume über einen definierten Mounting-Punkt und Namen verfügt. Vergewissern Sie sich, dass Sie in Ihrer Konfiguration in AWS OpsWorks Stacks den richtigen Mountpunkt angegeben haben, bevor Sie die Instance neu starten.

EBS-Volume wird nach einem Neustart nicht wieder zugeordnet

Problem: Sie verwenden die Amazon EC2 EC2-Konsole, um ein Amazon EBS-Volume an eine Instance anzuhängen, aber wenn Sie die Instance neu starten, ist das Volume nicht mehr angehängt.

Ursache: AWS OpsWorks Stacks können nur die Amazon EBS-Volumes wieder anhängen, die ihm bekannt sind. Diese sind auf Folgendes beschränkt:

- Volumes, die von Stacks erstellt wurden. AWS OpsWorks
- Die Volumes von Ihrem Konto wurden explizit über die Seite Resources (Ressourcen) für einen Stack registriert.

Lösung: Verwalten Sie Ihre Amazon EBS-Volumes nur mithilfe der AWS OpsWorks Stacks-Konsole, API oder CLI. Wenn Sie eines der Amazon EBS-Volumes Ihres Kontos mit einem Stack verwenden möchten, verwenden Sie die Ressourcenseite des Stacks, um das Volume zu registrieren und es an eine Instance anzuhängen. Weitere Informationen finden Sie unter Ressourcenmanagement.

AWS OpsWorks Stacks-Sicherheitsgruppen können nicht gelöscht werden

Problem: Nachdem Sie einen Stack gelöscht haben, bleiben eine Reihe von AWS OpsWorks Stacks-Sicherheitsgruppen übrig, die nicht gelöscht werden können.

Ursache: Die Sicherheitsgruppen müssen in einer bestimmten Reihenfolge gelöscht werden.

Lösung: Stellen Sie zunächst sicher, dass die Sicherheitsgruppen von keiner Instance verwendet werden. Anschließend löschen Sie die folgenden Sicherheitsgruppen, sofern vorhanden, in der folgenden Reihenfolge:

1. AWS OpsWorks - Blank-Server
2. OpsWorksAWS-Monitoring-Master-Server
3. OpsWorksAWS-DB-Master-Server
4. OpsWorksAWS-Memcached-Server
5. OpsWorksAWS-Benutzerdefinierter Server
6. OpsWorksAWS-NodeJS-App-Server
7. OpsWorksAWS-PHP-App-Server
8. OpsWorksAWS-Rails-App-Server
9. OpsWorksAWS-Webserver

10. AWS- OpsWorks -Standardserver

11. OpsWorksAWS-LB-Server

Versehentlich wurde eine Stacks-Sicherheitsgruppe AWS OpsWorks gelöscht

Problem: Sie haben eine der AWS OpsWorks Stacks-Sicherheitsgruppen gelöscht und müssen sie neu erstellen.

Ursache: Diese Sicherheitsgruppen werden meist aus Versehen gelöscht.

Lösung: Die neu erstellte Gruppe muss eine exakte Kopie des Originals einschließlich derselben Groß-/Kleinschreibung des Gruppennamens sein. Anstatt die Gruppe manuell neu zu erstellen, wird empfohlen, diese Schritte von AWS OpsWorks Stacks ausführen zu lassen. Erstellen Sie einfach einen neuen Stack in derselben AWS-Region — und VPC, falls vorhanden — und AWS OpsWorks Stacks erstellt automatisch alle integrierten Sicherheitsgruppen neu, einschließlich der gelöschten. Anschließend können Sie den Stack löschen, wenn Sie keine weitere Verwendung dafür haben. Die Sicherheitsgruppen bleiben erhalten.

Chef-Protokoll wird plötzlich beendet

Problem: Ein Chef-Protokoll wird plötzlich beendet und aus dem Ende des Protokolls geht nicht hervor, ob es sich um eine erfolgreiche Ausführung handelt. Es wird auch keine Ausnahme bzw. kein Stacktrace angezeigt.

Ursache: Dieses Verhalten wird in der Regel durch zu wenig Speicher verursacht.

Lösung: Erstellen Sie eine größere Instance und verwenden Sie den Agent-CLI-Befehl `run_command`, um die Rezepte erneut auszuführen. Weitere Informationen finden Sie unter Ausführen von Rezepten.

Rezeptbuch wird nicht aktualisiert

Problem: Sie haben Ihre Rezeptbücher aktualisiert, aber auf den Instances des Stacks werden immer noch die alten Rezepte ausgeführt.

Ursache: AWS OpsWorks Stacks speichert Kochbücher auf jeder Instanz im Cache und führt Rezepte aus dem Cache aus, nicht aus dem Repository. Wenn Sie eine neue Instanz starten, lädt AWS OpsWorks Stacks Ihre Kochbücher aus dem Repository in den Cache der Instanz herunter. Wenn Sie aber Ihre benutzerdefinierten Rezeptbücher häufig ändern, werden die Online-Instance-Caches von AWS OpsWorks Stacks nicht automatisch aktualisiert.

Lösung: Führen Sie den Befehl Update Cookbooks stack aus, um AWS OpsWorks Stacks explizit anzuweisen, die Kochbuch-Caches Ihrer Online-Instanzen zu aktualisieren.

Instances bleiben im Startstatus hängen

Problem: Wenn Sie eine Instance starten oder diese von der automatischen Reparatur automatisch neu gestartet wird, bleibt der Startvorgang im Status `booting` stehen.

Ursache: Ein möglicher Grund für dieses Problem ist die VPC-Konfiguration, einschließlich einer Standard-VPC. Instances müssen immer in der Lage sein, mit dem AWS OpsWorks Stacks-Service, Amazon S3 und den Paket-, Kochbuch- und App-Repositorys zu kommunizieren. Wenn Sie beispielsweise ein Standard-Gateway aus einer Standard-VPC entfernen, verlieren die Instances ihre Verbindung zum AWS OpsWorks Stacks-Service. Da AWS OpsWorks Stacks nicht mehr mit dem Instanzagenten kommunizieren kann, behandelt es die Instance als ausgefallen und repariert sie auto. Ohne eine Verbindung kann AWS OpsWorks Stacks jedoch keinen Instanzagenten auf der geheilten Instanz installieren. Ohne einen Agenten kann AWS OpsWorks Stacks die Setup-Rezepte auf der Instanz nicht ausführen, sodass der Startvorgang nicht über den Status „Booting“ hinaus fortgesetzt werden kann.

Lösung: Ändern Sie die VPC-Konfiguration so, dass Instances über die erforderliche Konnektivität verfügen.

Instances starten unerwartet neu

Problem: Eine gestoppte Instance startet unerwartet neu.

Ursache 1: Wenn Sie die auto Heilung für Ihre Instances aktiviert haben, führt AWS OpsWorks Stacks regelmäßig eine Zustandsprüfung der zugehörigen Amazon EC2 EC2-Instances durch und startet alle fehlerhaften neu. Wenn Sie eine von AWS OpsWorks Stacks verwaltete Instance mithilfe der Amazon EC2 EC2-Konsole, API oder CLI stoppen oder beenden, wird AWS OpsWorks Stacks nicht benachrichtigt. Stattdessen wird die gestoppte Instance als fehlerhaft erkannt und automatisch neu gestartet.

Lösung: Verwalten Sie Ihre Instances nur über die Konsole, die API oder die CLI von AWS OpsWorks Stacks. Wenn Sie AWS OpsWorks Stacks verwenden, um eine Instance zu beenden oder zu löschen, wird sie nicht neu gestartet. Weitere Informationen finden Sie unter Manuelles Starten, Beenden und Neustarten von 24/7-Instances und AWS OpsWorks Stacks-Instances löschen.

Ursache 2: Instances können aus verschiedenen Gründen fehlerhaft sein. Wenn Sie Auto Healing aktiviert haben, startet AWS OpsWorks Stacks ausgefallene Instances automatisch neu.

Lösung: Dies ist ein normaler Vorgang. Sie müssen nichts tun, es sei denn, Sie möchten nicht, dass AWS OpsWorks Stacks ausgefallene Instances neu startet. In diesem Fall sollten Sie die automatische Reparatur deaktivieren.

`opsworks-agent`-Prozesse werden auf Instances ausgeführt

Problem: Auf den Instances werden mehrere `opsworks-agent`-Prozesse ausgeführt. Beispielsweise:

```
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
```

Ursache: Dies sind reguläre Prozesse, die für eine normale Agent-Ausführung erforderlich sind. Sie führen Aufgaben wie das Verarbeiten von Bereitstellungen und das Senden von Keepalive-Nachrichten an den Service aus.

Lösung: Das ist ein normales Verhalten. Beenden Sie diese Prozesse nicht, denn das beeinträchtigt die Agent-Ausführung.

Unerwartete "execute_recipes"-Befehle

Problem: Der Bereich Logs (Protokolle) auf der Instance-Detailseite enthält unerwartete `execute_recipes`-Befehle. Unerwartete `execute_recipes`-Befehle können auch auf den Seiten Stack und Deployments (Bereitstellungen) angezeigt werden.

Ursache: Dieses Problem wird meist von geänderten Berechtigungen verursacht. Wenn Sie die SSH- oder Sudo-Berechtigungen eines Benutzers oder einer Gruppe ändern, wird AWS OpsWorks Stacks ausgeführt, um die Instanzen des Stacks `execute_recipes` zu aktualisieren, und löst außerdem ein Configure-Ereignis aus. Die `execute_recipes`-Befehle können außerdem auch dadurch verursacht werden, dass AWS OpsWorks Stacks den Instance-Agent aktualisiert.

Lösung: Hierbei handelt es sich um einen normalen Vorgang, es sind keine Schritte erforderlich.

Um die von einem `execute_recipes`-Befehl ausgeführten Aktionen anzuzeigen, klicken Sie auf der Seite Deployments (Bereitstellungen) auf den Zeitstempel des Befehls. Dann wird die Detailseite des Befehls mit den wichtigsten ausgeführten Rezepten angezeigt. Beispielsweise ist die folgende

Detailseite für einen `execute_recipes`-Befehl, der `ssh_users` zur Aktualisierung der SSH-Berechtigungen ausgeführt hat.

Zum Anzeigen aller Details klicken Sie auf show (Anzeigen) in der Spalte Log (Protokoll) des Befehls. Damit zeigen Sie das zugehörige Chef-Protokoll an. Suchen Sie im Protokoll nach. **`Run List`** AWS OpsWorks Rezepte für die Wartung von Stacks finden Sie unter`OpsWorks Custom Run List`. Beispielsweise sieht die Ausführungsliste für den im vorigen Screenshot abgebildeten `execute_recipes`-Befehl, die alle Rezepte anzeigt, die diesem Befehl zugeordnet sind, wie folgt aus.

```
[2014-02-21T17:16:30+00:00] INFO: OpsWorks Custom Run List:
 ["opsworks_stack_state_sync",
  "ssh_users", "test_suite", "opsworks_cleanup"]
```

Fehlerbehebung bei der Instance-Registrierung

Dieser Abschnitt enthält Lösungen für einige bekannte Fehler bei der Instance-Registrierung.

> **Note**
>
> Falls bei der Registrierung Probleme auftreten, führen Sie `register` mit dem Argument `--debug` aus, das zusätzliche Debugging-Informationen bietet.

Themen

- EC2User ist nicht berechtigt zur Ausführung von: ...
- Anmeldeinformationen müssen sich auf gültige Region beziehen

EC2User ist nicht berechtigt zur Ausführung von: ...

Problem: Ein `register`-Befehl gibt Folgendes zurück:

```
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
```

Ursache: Der `register` Befehl wird mit Anmeldeinformationen ausgeführt, die nicht die erforderlichen Berechtigungen gewähren. Die Richtlinie für den Benutzer muss unter anderen die Aktion `iam:CreateGroup` zulassen.

Lösung: Stellen Sie für `register` IAM-Benutzeranmeldeinformationen mit entsprechenden Berechtigungen bereit. Weitere Informationen finden Sie unter Installieren und Konfigurieren der AWS CLI.

Anmeldeinformationen müssen sich auf gültige Region beziehen

Problem: Ein `register`-Befehl gibt Folgendes zurück:

```
A client error (InvalidSignatureException) occurred when calling the
DescribeStacks operation: Credential should be scoped to a valid region, not 'cn-
north-1'.
```

Ursache: Bei der Region im Befehl muss es sich um eine gültige AWS OpsWorks Stacks-Region handeln. Eine Liste der unterstützten Regionen finden Sie unter Unterstützung von Regionen. Dieser Fehler tritt in der Regel aus einem der folgenden Gründe auf:

- Der Stack befindet sich in einer anderen Region und Sie haben die Stack-Region dem `--region`-Argument im Befehl zugeordnet.

  Sie müssen keine Stack-Region angeben. AWS OpsWorks Stacks bestimmt sie automatisch anhand der Stack-ID.
- Sie haben das `--region`-Argument ausgelassen, das implizit die Standardregion angibt, die jedoch von AWS OpsWorks Stacks nicht unterstützt wird.

Lösung: Geben Sie explizit eine unterstützte AWS OpsWorks Stacks-Region `--region` an oder bearbeiten Sie Ihre AWS CLI `config` Datei, um die Standardregion in eine unterstützte AWS

OpsWorks Stacks-Region zu ändern. Weitere Informationen finden Sie unter Konfigurieren der AWS-Befehlszeilenschnittstelle.

AWS OpsWorks Stacks Agent CLI

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

> **Note**
>
> Diese Funktion ist nur für Linux-Instances verfügbar.

Der Agent, den AWS OpsWorks Stacks auf jeder Instanz installiert, stellt eine Befehlszeilenschnittstelle (CLI) zur Verfügung. Wenn Sie SSH für die Anmeldung bei der Instance verwenden, können Sie die CLI für Folgendes benutzen:

- Auf die Protokolldateien der Chef-Ausführungen zugreifen.
- Greifen Sie auf AWS OpsWorks Stacks-Befehle zu.
- Chef-Rezepte manuell ausführen.
- Instance-Berichte anzeigen lassen.
- Agent-Reporte anzeigen lassen.
- Einen begrenzten Satz von Attributen der Stack-Konfiguration und der Bereitstellung ansehen.

> **Important**
>
> Sie können Agenten-CLI-Befehle nur als root-Benutzer ausführen oder durch die Benutzung von `sudo`.

Die grundlegende Befehlssyntax ist:

```
sudo opsworks-agent-cli [--help] [command [activity] [date]]
```

Die vier Argumente sind wie folgt:

help

(Optional) Zeigt eine kurze Zusammenfassung der verfügbaren Befehle an, wenn sie einzeln verwendet werden. Bei der Verwendung mit einem Befehl zeigt `help` eine Beschreibung des Befehls an.

command

(Optional) Der Agenten-CLI-Befehl, der auf eine der folgenden Einstellungen gesetzt sein muss:

- agent_report
- get_json
- instance_report
- list_commands
- run_command
- show_log
- stack_state

Aktivität

(Optional) Wird als ein Argument mit einigen Befehlen benutzt, um eine bestimmte AWS OpsWorks Stacks-Aktivität anzugeben: `setup`, `configure`, `deploy`, `undeploy`, `start`, `stop` oder `restart`.

date

(Optional) Wird als Argument mit einigen Befehlen benutzt, um eine bestimmte AWS OpsWorks Stacks-Befehlsausführung anzugeben. Geben Sie die Befehlsausführung an, indem Sie das Datum auf den Zeitstempel der Ausführung des Befehls im Format *`yyyy-mm-ddTHH:MM:SS`* setzen, einschließlich der einfachen Anführungszeichen. Verwenden Sie zum Beispiel für 10:31:55 am Dienstag, den 5. Februar 2013: `'2013-02-05T10:31:55'`. Um festzustellen, wann ein bestimmter AWS OpsWorks Stacks-Befehl ausgeführt wurde, führen Sie den Befehl aus. list_commands

> **Note**
>
> Wenn der Agent dieselbe AWS OpsWorks Stacks-Aktivität mehrmals ausgeführt hat, können Sie eine bestimmte Ausführung auswählen, indem Sie sowohl die Aktivität als auch die Uhrzeit angeben, zu der sie ausgeführt wurde. Wenn Sie eine Aktivität angeben und die Zeit weglassen, wird der Agenten-CLI-Befehl anhand der letzten Ausführung dieser Aktivität angewendet. Wenn Sie beide Argumente weglassen, wird der Agenten-CLI-Befehl anhand der Aktivität ausgeführt.

In den folgenden Abschnitten werden die Befehle und die zugehörigen Argumente beschrieben. Der Kürze halber wird bei den Syntaxabschnitten auf die Option `--help` verzichtet, da diese mit allen Befehlen genutzt werden kann.

Themen

agent_report

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Gibt einen Agent-Report zurück.

```
sudo opsworks-agent-cli agent_report
```

Das folgende Ausgabebeispiel stammt von einer Instance, die zuletzt eine Konfigurationsaktivität durchgeführt hat.

```
$ 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**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Gibt Informationen über eine Chef-Ausführung als JSON-Objekt zurück.

```
sudo opsworks-agent-cli get_json [activity] [date] [-i | --internal | --no-i | --no-
internal]
```

Standardmäßig zeigt `get_json` die vom Kunden bereitgestellten Informationen für die neueste Chef-Ausführung an. Verwenden Sie die folgenden Optionen, um eine bestimmte Gruppe von Informationen anzugeben.

Aktivität

Zeigt Informationen für die Chef-Ausführung im Zusammenhang mit der zuletzt angegebenen Aktivität an. Um eine Liste von gültigen Aktivitäten zu erhalten, führen Sie list_commands aus.

date

Zeigt Informationen für die Chef-Ausführung im Zusammenhang mit der Aktivität an, die für den festgelegten Zeitstempel ausgeführt wurde. Um eine Liste von gültigen Zeitstempeln zu erhalten, führen Sie list_commands aus.

-i, --internal

Zeigt Informationen an, die AWS OpsWorks Stacks intern für den Chef-Lauf verwendet.

--no-i, --no-internal

Zeigt explizit vom Kunden bereitgestellte Informationen für die Chef-Ausführungen an. Dies ist der Standardwert, wenn nicht anders angegeben.

> **Note**
>
> Für Chef 12 Linux-Instances wird die Ausführung dieses Befehls gültige Informationen zurückgeben, wie die Attribute der Instance-Stack-Konfiguration und Bereitstellungsattribute. Um umfassendere Informationen zu erhalten, verweisen Sie jedoch auf die Chef-Datenpakete, die AWS OpsWorks Stacks auf der Instanz erstellt. Weitere Informationen hierzu finden Sie unter AWS OpsWorks Referenz für Stacks Data Bag.

Das folgende Beispiel einer Ausgabe zeigt die vom Kunden bereitgestellten Informationen für die neueste Chef-Ausführung für die zuletzt durchgeführte Konfigurationsaktivität.

```
$ sudo opsworks-agent-cli get_json configure

{
  "run_list": [
    "recipe[opsworks_cookbook_demo::configure]"
  ]
}
```

Das folgende Ausgabebeispiel zeigt Informationen, die AWS OpsWorks Stacks intern für den Chef-Lauf verwendet, der für den angegebenen Zeitstempel ausgeführt wird.

```
$ 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/opsworks-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**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Gibt einen erweiterten Instance-Report zurück.

```
sudo opsworks-agent-cli instance_report
```

Das folgende Ausgabebeispiel stammt von einer Instance.

```
$ 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**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Notiert die Zeiten für jede Aktivität, die in dieser Instance ausgeführt wurde. Sie können diese Zeiten für andere Agenten-CLI-Befehle benutzen, um eine bestimmte Ausführung festzulegen.

```
sudo opsworks-agent-cli list_commands  [activity] [date]
```

Das folgende Ausgabebeispiel stammt von einer Instance, die die Aktivitäten Konfigurieren, Einrichten und Aktualisieren von benutzerdefinierten Rezeptbüchern ausgeführt hat.

```
$ 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**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Führt einen AWS OpsWorks Stacks-Befehl aus, bei dem es sich um eine JSON-Datei handelt, die eine Chef-Run-Liste enthält, die die Informationen enthält, die zur Ausführung einer AWS OpsWorks Stacks-Aktivität (Einrichtung, Konfiguration, Bereitstellung usw.) erforderlich sind. Der `run_command`-Befehl generiert einen Protokolleintrag, den Sie durch das Ausführen von `show_log` aufrufen können. Diese Option ist nur für Entwicklungszwecke vorgesehen, sodass AWS OpsWorks Stacks keine Änderungen verfolgt.

```
sudo opsworks-agent-cli run_command [activity] [date] [/path/to/valid/json.file]
```

`run_command`Führt standardmäßig den neuesten AWS OpsWorks Stacks-Befehl aus. Verwenden Sie die folgenden Optionen, um einen bestimmten Befehl anzugeben.

Aktivität

Führt einen angegebenen AWS OpsWorks Stacks-Befehl aus:`setup`,`configure`,,`deploy`,`undeploy`, `startstop`, oder. `restart`

date

Führen Sie den OpsWorks AWS-Befehl aus, der zum angegebenen Zeitstempel ausgeführt wurde. Um eine Liste von gültigen Zeitstempeln zu erhalten, führen Sie `list_commands` aus.

file

Führen Sie den angegebenen JSON-Datei-Befehl aus. Um den Dateipfad eines Befehls zu erhalten, führen Sie `get_json` aus.

Das folgende Ausgabebeispiel stammt von einer Instance und führt den Konfigurationsbefehl aus.

```
$ 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**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Gibt die Protokolldatei eines Befehls zurück.

```
sudo opsworks-agent-cli show_log [activity] [date]
```

Standardmäßig zeigt `show_log` die aktuelle Protokolldatei an. Verwenden Sie die folgenden Optionen, um einen bestimmten Befehl anzugeben.

Aktivität

Zeigen Sie die Protokolldatei der angegebenen Aktivität an.

date

Zeigen Sie die Protokolldateien für die Aktivität an, die mit zu dem angegebenen Zeitstempel ausgeführt wurde. Um eine Liste von gültigen Zeitstempeln zu erhalten, führen Sie list_commands aus.

Das folgende Ausgabebeispiel zeigt das neueste Protokoll an.

```
$ 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**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Zeigt Informationen an, die AWS OpsWorks Stacks intern für den letzten Chef-Lauf verwendet.

```
opsworks-agent-cli stack_state
```

> **Note**
>
> Für Chef 12 Linux-Instances wird die Ausführung dieses Befehls gültige Informationen zurückgeben, wie die Attribute der Instance-Stack-Konfiguration und Bereitstellungsattribute. Um umfassendere Informationen zu erhalten, verweisen Sie jedoch auf die Chef-Datenpakete, die AWS OpsWorks Stacks auf der Instanz erstellt. Weitere Informationen hierzu finden Sie unter AWS OpsWorks Referenz für Stacks Data Bag.

Das folgende Ausgabebeispiel stammt von einer Instance.

```
$ 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/opsworks-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 Referenz für Stacks Data Bag

Important

Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir

empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

AWS OpsWorks Stacks stellt eine Vielzahl von Einstellungen für Rezepte als Inhalt der Chef-Datentüte zur Verfügung. Diese Referenz führt die Data Bag-Inhalte auf.

Ein Data Bag ist ein Chef-Konzept. Bei einem Data Bag handelt es sich um eine globale Variable, die in Form von JSON-Daten auf einer Instance gespeichert wird. Auf die JSON-Daten wird mit Chef zugegriffen. In einer Datentasche können beispielsweise globale Variablen wie die Quell-URL einer App, der Hostname der Instanz und die VPC-ID des zugehörigen Stacks gespeichert werden. AWS OpsWorks Stacks speichert seine Datentaschen auf den Instanzen jedes Stacks. Auf Linux-Instances speichert AWS OpsWorks Stacks Datentaschen im `/var/chef/runs/`*`run-ID`*`/data_bags` Verzeichnis. Auf Windows-Instances werden Data Bags im Verzeichnis *`drive`*`:\chef\runs\`*`run-id`*`\data_bags` gespeichert. In beiden Fällen ist *`Run-ID eine eindeutige ID`*, die AWS OpsWorks Stacks jedem Chef-Lauf auf einer Instanz zuweist. Diese Verzeichnisse enthalten mehrere Data Bags (Unterverzeichnisse). Jedes Data Bag enthält null oder mehr Data Bag-Elemente. Das sind Dateien im JSON-Format, die Data Bag-Inhalte enthalten.

> **Note**
>
> AWS OpsWorks Stacks unterstützt keine verschlüsselten Datenbeutel. Um vertrauliche Daten in verschlüsselter Form zu speichern, wie z. B. Passwörter oder Zertifikate, empfehlen wir, diese in einem privaten S3-Bucket zu speichern. Anschließend können Sie ein benutzerdefiniertes Rezept erstellen, das zum Abrufen der Daten das Amazon SDK für Ruby verwendet. Ein Beispiel finden Sie unter Verwenden des -SDK for Ruby.

Ein Data Bag kann folgende Inhalte enthalten:

- Zeichenfolgen nach der Ruby-Standardsyntax, mit einfachen oder doppelten Anführungszeichen (für Zeichenfolgen mit bestimmten Sonderzeichen müssen doppelte Anführungszeichen gesetzt werden). Weitere Informationen finden Sie auf der Website der Ruby-Dokumentation.
- Boolesche Werte, also entweder `true` oder `false` (ohne Anführungszeichen).
- Ziffern in Form von Ganzzahlen oder Dezimalzahlen wie z. B. `4` oder `2.5` (ohne Anführungszeichen).

- Listen im Format von CSV-Werten in eckigen Klammern (ohne Anführungszeichen), wie z. B. `[ '80', '443' ]`
- JSON-Objekte mit zusätzlichen Data Bag-Inhalten, wie z. B. `"my-app": {"elastic_ip": null,...}`.

Chef-Rezepte können mithilfe der Chef-Suchfunktion oder direkt auf Data Bags, Data Bag-Elemente und Data Bag-Inhalte zugreifen. Nachfolgend werden beide Zugriffsmethoden beschrieben (obwohl die Chef-Suchfunktion bevorzugt wird).

Um über die Chef-Suche auf eine Datentasche zuzugreifen, verwenden Sie die Suchmethode und geben Sie den gewünschten Suchindex an. AWS OpsWorks Stacks bietet die folgenden Suchindizes:

- aws_opsworks_app bildet die bereitgestellten Apps für einen Stack ab.
- aws_opsworks_command bildet die Befehle ab, die für einen Stack ausgeführt wurden.
- aws_opsworks_ecs_cluster, was eine Reihe von Amazon Elastic Container Service (Amazon ECS) -Cluster-Instances für einen Stack darstellt.
- aws_opsworks_elastic_load_balancer, was eine Reihe von Elastic Load Balancing-Load Balancing-Load Balancern für einen Stack darstellt.
- aws_opsworks_instance bildet die Instances für einen Stack ab.
- aws_opsworks_layer bildet die Layer für einen Stack ab.
- aws_opsworks_rds_db_instance, was eine Reihe von Amazon Relational Database Service (Amazon RDS) -Instances für einen Stack darstellt.
- aws_opsworks_stack bildet einen Stack ab.
- aws_opsworks_user bildet die Benutzer für einen Stack ab.

Wenn Sie den Namen des Suchindex kennen, können Sie auf die Data Bag-Inhalte dieses Suchindex zugreifen. Beispielsweise wird von folgendem Rezeptcode der Suchindex `aws_opsworks_app` verwendet, um die Inhalte des ersten Data Bag-Elements (die erste JSON-Datei) aus dem Data Bag `aws_opsworks_app` (Verzeichnis `aws_opsworks_app`) abzurufen. Der Code schreibt dann zwei Nachrichten in das Chef-Protokoll, die eine enthält die Data Bag-Inhalte mit dem App-Kurznamen (eine Zeichenfolge in der JSON-Datei), die andere enthält die Data Bag-Inhalte mit der App-Quell-URL (eine weitere Zeichenfolge in der JSON-Datei):

```
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']}' **********")
```

Hier geben `['shortname']` und `['app_source']['url']` die folgenden Data Bag-Inhalte in der zugehörigen JSON-Datei an:

```
{
  ...
  "shortname": "mylinuxdemoapp",
  ...
  "app_source": {
    ...
    "url": "https://s3.amazonaws.com/opsworks-demo-assets/opsworks-linux-demo-
nodejs.tar.gz",
  },
  ...
}
```

Eine Liste der suchbaren Data Bag-Inhalte finden Sie im Thema "Referenz" in diesem Abschnitt.

Sie können die Data Bag-Elemente in einem Data Bag auch schrittweise durchlaufen. Beispielsweise ist der folgende Rezeptcode identisch mit dem vorherigen Beispiel. Hier werden die einzelnen Data Bag-Elemente im Data Bag schrittweise durchlaufen, wenn mehr als ein Data Bag-Element vorhanden ist:

```
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
```

Wenn Sie wissen, dass bestimmte Data Bag-Inhalte vorhanden sind, können Sie mit der folgenden Syntax nach dem entsprechenden Data Bag-Element suchen:

```
search("search_index", "key:value").first
```

Beispielsweise wird von folgendem Rezeptcode der Suchindex `aws_opsworks_app` verwendet, um nach dem Data Bag-Element zu suchen, das den App-Kurznamen `mylinuxdemoapp` enthält. Anschließend wird unter Verwendung der Data Bag-Elementinhalte eine Nachricht mit dem Kurznamen und der Quell-URL der betreffenden App in das Chef-Protokoll geschrieben:

```
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']}' **********")
```

Nur bei dem Suchindex `aws_opsworks_instance` können Sie mit `self:true` die Instance angeben, auf der das Rezept ausgeführt wird. Der folgende Rezeptcode verwendet den Inhalt des entsprechenden Datenbeutelelements, um eine Nachricht mit der von Stacks generierten ID und dem Betriebssystem der entsprechenden Instance in das Chef-Protokoll zu schreiben: 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']}' **********")
```

Sie können auch direkt auf Data Bags, Data Bag-Elemente und Data Bag-Inhalte zugreifen, ohne die Chef-Suchfunktion zu nutzen. Verwenden Sie dazu die Methoden data_bag und data_bag_item für den Zugriff auf Data Bags bzw. Data Bag-Elemente. Beispielsweise werden mit dem folgenden Rezeptcode die gleichen Schritte ausgeführt wie in den vorherigen Beispielen, nur wird hier direkt auf ein einzelnes Data Bag-Element zugegriffen (bzw. auf mehrere, sofern vorhanden):

```
# 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
```

Von diesen beiden Methoden wird die Verwendung der Chef-Suchfunktion empfohlen. In allen zugehörigen Beispielen in diesem Handbuch wird diese Methode herangezogen.

Themen

- Data Bag für Apps (aws_opsworks_app)
- Data Bag für Befehle (aws_opsworks_command)
- Amazon ECS-Cluster-Datentasche (aws_opsworks_ecs_cluster)

- Elastic Load Balancing Balancing-Datentasche (aws_opsworks_elastic_load_balancer)
- Data Bag für Instances (aws_opsworks_instance)
- Data Bag für Layer (aws_opsworks_layer)
- Amazon RDS-Datentasche (aws_opsworks_rds_db_instance)
- Data Bag für Stacks (aws_opsworks_stack)
- Data Bag für Benutzer (aws_opsworks_user)

Data Bag für Apps (aws_opsworks_app)

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Enthält bei einem Deploy-Ereignis oder dem Stack-Befehl "Execute Recipes" die App-Einstellungen.

Das folgende Beispiel zeigt, wie Sie mit der Chef-Suchfunktion ein einzelnes Data Bag-Element (und anschließend mehrere Data Bag-Elemente) durchsuchen und Nachrichten mit den Kurznamen und den Quell-URLs der Apps ins Chef-Protokoll schreiben:

```
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

deploy	Attribute	domains
enable_ssl	Umgebung	Name
shortname	ssl_configuration	Typ

app_id

Die App-ID (Zeichenfolge). Eine GUID zur Identifizierung der Anwendung.

app_source

Eine Reihe von Inhalten, die die Informationen spezifizieren, die AWS OpsWorks Stacks für die Bereitstellung der App aus seinem Quellcodeverwaltungs-Repository verwendet. Die Inhalte sind abhängig vom Repository-Typ.

password

Das Passwort für private Repositorys und `"null"` für öffentliche Repositorys (Zeichenfolge). Bei privaten S3-Buckets sind diese Inhalte auf den geheimen Schlüssel festgelegt.

Änderung

Falls das Repository über mehrere Branches verfügt, geben die Inhalte den Branch oder die Version der App an, z. B. `"version1"` (Zeichenfolge). Andernfalls lautet der Wert `"null"`.

ssh_key

Ein SSH-Bereitstellungsschlüssel für den Zugriff auf private Git-Repositorys und `"null"` für öffentliche Repositorys (Zeichenfolge).

Typ

Der Quellspeicherort der App (Zeichenfolge). Gültige Werte sind:

- `"archive"`
- `"git"`
- `"other"`
- `"s3"`

URL

Gibt an, wo sich die App-Quelle befindet (Zeichenfolge).

user

Der Benutzername für private Repositorys und `"null"` für öffentliche Repositorys (Zeichenfolge). Bei privaten S3-Buckets sind die Inhalte auf den Zugriffsschlüssel festgelegt.

Attribute

Diese Inhalte beschreiben die Verzeichnisstruktur und die Inhalte der App.

document_root

Das Stammverzeichnis der Dokumentstruktur. Definiert den Pfad zum Dokumentstamm oder zur App-Startseite wie z. B. `home_html` relativ zum Bereitstellungsverzeichnis. Wenn dieses Attribut nicht angegeben wird, ist `public` der Standardwert für "document_root". Der Wert von `document_root` muss als erstes Zeichen `a-z`, `A-Z`, `0-9`, `_` (Unterstrich) oder `-` (Bindestrich) aufweisen.

data_sources

Diese Informationen sind für die Verbindung zur App-Datenbank erforderlich. Wenn der App eine Datenbankschicht angehängt ist, weist AWS OpsWorks Stacks diesem Inhalt automatisch die entsprechenden Werte zu.

Der Wert von "data_sources" ist ein Array; und auf Arrays wird per Integral-Offset (und nicht über Schlüssel) zugegriffen. Verwenden Sie beispielsweise für den Zugriff auf die erste App-Datenquelle `app[:data_sources][0][:type]`.

database_name

Der Datenbankname – in der Regel der App-Kurzname (Zeichenfolge).

Typ

Der Datenbank-Instance-Typ – in der Regel `"RdsDbInstance"` (Zeichenfolge).

arn

Der Amazon-Ressourcenname (ARN) der Datenbank-Instance (Zeichenfolge).

deploy

Gibt an, ob die App bereitgestellt werden soll (Boolescher Wert). Für Apps, die in einem Deploy-Lebenszyklusereignis bereitgestellt werden sollen, gilt der Wert `true`. Bei einem Setup-Lebenszyklusereignis haben diese Inhalte den Wert `true` für alle Apps. Um zu bestimmen, welche Apps auf einer Instance bereitgestellt werden sollen, prüfen Sie die Layer, denen die Instance angehört.

domains

Eine Liste der App-Domänen (Liste aus Zeichenfolgen).

enable_ssl

Gibt an, ob SSL-Unterstützung aktiviert ist (Boolescher Wert).

Umgebung

Eine Sammlung von benutzerdefinierten Umgebungsvariablen, die für die Anwendung definiert wurden. Weitere Informationen zur Definition von Umgebungsvariablen für eine App finden Sie unter Hinzufügen von Apps. Jeder Inhaltsname wird auf einen Umgebungsvariablennamen und der entsprechende Wert auf den Variablenwert festgelegt.

Name

Der App-Name, der für die Anzeige verwendet wird (Zeichenfolge).

shortname

Der Kurzname der App, der von AWS OpsWorks Stacks aus dem Namen (Zeichenfolge) generiert wird. Der Kurzname wird intern und von Rezepten verwendet. Zudem wird er als Name des Verzeichnisses genutzt, in dem die App-Dateien installiert sind.

ssl_configuration

Zertifikat

Sofern die SSL-Unterstützung aktiviert ist, wird hier das SSL-Zertifikat der App angegeben. Andernfalls lautet der Wert `"null"` (Zeichenfolge).

chain

Sofern SSL aktiviert ist, werden hier Inhalte für den Zertifizierungsstellenschlüssel des Zwischenzertifikats oder die Clientauthentifizierung angegeben (Zeichenfolge).

private_key

Sofern die SSL-Unterstützung aktiviert ist, wird hier der private SSL-Schlüssel für die App angegeben. Andernfalls lautet der Wert `"null"` (Zeichenfolge).

Typ

Der App-Typ, der bei Chef 12 Linux-Stacks und Chef 12.2 Windows-Stacks immer auf `"other"` festgelegt ist (Zeichenfolge).

Data Bag für Befehle (aws_opsworks_command)

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Stellt Einstellungen für einen Befehl dar, den AWS OpsWorks Stacks auf einer oder mehreren Instanzen ausführt.

Das folgende Beispiel zeigt, wie Sie mit der Chef-Suchfunktion ein einzelnes Data Bag-Element (und anschließend mehrere Data Bag-Elemente) durchsuchen und Nachrichten mit den Typen und den Sendezeitpunkten der Befehle ins Chef-Protokoll schreiben:

```
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	Typ

args

Argumente für den Befehl (Zeichenfolge).

command_id

Die zufällige eindeutige Kennung des Befehls, die von AWS OpsWorks Stacks (Zeichenfolge) zugewiesen wurde.

iam_user_arn

Sofern der Befehl vom Kunden erstellt wird, ist dies der Amazon-Ressourcenname des Benutzers, der den Befehl erstellt hat (Zeichenfolge).

instance_id

Die ID der Instance, auf der dieser Befehl ausgeführt wurde (Zeichenfolge).

sent_at

Der Zeitstempel, zu dem AWS OpsWorks Stacks den Befehl ausgeführt hat (Zeichenfolge).

Typ

Der Typ des Befehls (Zeichenfolge). Gültige Werte sind:

- `"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"`

Amazon ECS-Cluster-Datentasche (aws_opsworks_ecs_cluster)

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Stellt die Einstellungen eines Amazon ECS-Clusters dar.

Das folgende Beispiel zeigt, wie Sie die Chef-Suche verwenden, um ein einzelnes Datenbeutelelement und dann mehrere Datenbeutelelemente zu durchsuchen, um Nachrichten mit den Namen der Amazon ECS-Cluster und den Amazon-Ressourcennamen (ARNs) in das Chef-Protokoll zu schreiben:

```
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

Der Amazon-Ressourcenname (ARN) des Clusters (Zeichenfolge).

ecs_cluster_name

Der Clustername (Zeichenfolge).

Elastic Load Balancing Balancing-Datentasche (aws_opsworks_elastic_load_balancer)

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Stellt die Einstellungen eines Elastic Load Balancing-Load Balancers dar.

Das folgende Beispiel zeigt, wie Sie die Chef-Suche verwenden, um ein einzelnes Datenbeutelelement und dann mehrere Datenbeutelelemente zu durchsuchen, um Nachrichten mit den Namen und DNS-Namen der Elastic Load Balancing Load Balancer in das Chef-Protokoll zu schreiben:

```
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

Der Load Balancer-Name (Zeichenfolge).

dns_name

Der DNS-Name des Load Balancers (Zeichenfolge).

layer_id

Die AWS OpsWorks Stacks-ID der Ebene, der der Load Balancer zugewiesen ist (Zeichenfolge).

Data Bag für Instances (aws_opsworks_instance)

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Enthält die Einstellungen für eine Instance.

Das folgende Beispiel zeigt, wie Sie mit der Chef-Suchfunktion ein einzelnes Data Bag-Element (und anschließend mehrere Data Bag-Elemente) durchsuchen und Nachrichten mit den Host-Namen und den IDs der Instances ins Chef-Protokoll schreiben:

```
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
```

Das folgende Beispiel zeigt verschiedene Möglichkeiten, die Chef-Suche zu verwenden, um mehrere Datenbeutelelemente zu durchsuchen, um das Datenbeutelelement zu finden, das die angegebene

Amazon EC2 EC2-Instance-ID enthält. Anschließend wird unter Verwendung der Data Bag-Elementinhalte eine Nachricht mit der öffentlichen IP-Adresse der betreffenden Instance in das Chef-Protokoll geschrieben:

```
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
```

Das folgende Beispiel zeigt, wie Sie mit der Chef-Suchfunktion und mit `self:true` nach dem Data Bag-Element suchen, das die Informationen zur Instance enthält, auf der das Rezept ausgeführt wird. Das Beispiel verwendet dann den Inhalt des AWS OpsWorks Datensackelements, um eine Nachricht mit der von Stacks generierten ID der entsprechenden Instanz und der öffentlichen IP-Adresse der Instance in das Chef-Protokoll zu schreiben:

```
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	Anwendung ansehen	auto_scaling_type
availability_zone	created_at	ebs_optimized
ec2_instance_id	elastic_ip	hostname
instance_id	instance_type	layer_ids
os	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

Die AMI (Amazon Machine Image)-ID der Instance (Zeichenfolge).

Anwendung ansehen

Die Instance-Architektur, die immer den Wert "x86_64" hat (Zeichenfolge).

auto_scaling_type

Der Skalierungstyp der Instance, entweder null, timer oder load (Zeichenfolge).

availability_zone

Die Availability Zone (AZ) der Instance, z. B. "us-west-2a" (Zeichenfolge).

created_at

Der Zeitpunkt der Instance-Erstellung, im UTC-Format "*yyyy*-*mm*-*ddd*T*hh*:*mm*:*ss*+*hh*:*mm*" (Zeichenfolge). Beispielsweise gibt der Wert "2013-10-01T08:35:22+00:00" den 10. Oktober 2013 um 8:35:22 Uhr ohne Zeitzonenabweichung an. Weitere Informationen finden Sie unter ISO 8601.

ebs_optimized

Gibt an, ob die Instance für EBS optimiert ist (Boolescher Wert).

ec2_instance_id

Die EC2-Instance-ID (Zeichenfolge).

elastic_ip

Die Elastic IP-Adresse; wird auf "null" festgelegt, falls die Instance keine Elastic IP-Adresse hat (Zeichenfolge).

hostname

Der Host-Name, z. B. "demo1" (Zeichenfolge)

instance_id

Die Instanz-ID, bei der es sich um eine von AWS OpsWorks Stacks generierte GUID handelt, die die Instanz eindeutig identifiziert (Zeichenfolge).

instance_type

Der Instance-Typ, z. B. `"c1.medium"` (Zeichenfolge)

layer_ids

Eine Liste der Instance-Layer, gekennzeichnet durch eindeutige IDs, z. B. `307ut64c-c7e4-40cc-52f0-67d5k1f9992c`.

os

Das Betriebssystem der Instance (Zeichenfolge). Gültige Werte sind:

- `"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

Der private DNS-Name (Zeichenfolge).

private_ip

Die private IP-Adresse (Zeichenfolge).

public_dns

Der öffentliche DNS-Name (Zeichenfolge).

public_ip

Die öffentliche IP-Adresse (Zeichenfolge).

root_device_type

Der Root-Gerätetyp (Zeichenfolge). Gültige Werte sind:

- `"ebs`
- `"instance-store"`

root_device_volume_id

Die Volume-ID des Root-Geräts (Zeichenfolge).

self

Hat den Wert `true`, sofern dieses Data Bag-Element Informationen zur Instance enthält, auf der das Rezept ausgeführt wird. Andernfalls lautet der Wert `false` (Boolescher Wert). Dieser Wert ist nur für Rezepte verfügbar, nicht über die Stacks-API. AWS OpsWorks

ssh_host_dsa_key_fingerprint

Diese kürzere Bytesequenz dient der Identifizierung des längeren öffentlichen DSA-Schlüssels (Zeichenfolge).

ssh_host_dsa_key_private

Der per DSA generierte private Schlüssel für die SSH-Authentifizierung an der Instance (Zeichenfolge).

ssh_host_dsa_key_public

Der per DSA generierte öffentliche Schlüssel für die SSH-Authentifizierung an der Instance (Zeichenfolge).

ssh_host_rsa_key_fingerprint

Diese kürzere Bytesequenz dient der Identifizierung des längeren öffentlichen RSA-Schlüssels (Zeichenfolge).

ssh_host_rsa_key_private

Der per RSA generierte private Schlüssel für die SSH-Authentifizierung an der Instance (Zeichenfolge).

ssh_host_rsa_key_public

Der per RSA generierte öffentliche Schlüssel für die SSH-Authentifizierung an der Instance (Zeichenfolge).

Status

Der Status der Instance (Zeichenfolge). Gültige Werte sind:

- `"requested"`
- `"booting"`
- `"running_setup"`
- `"online"`
- `"setup_failed"`
- `"start_failed"`
- `"terminating"`
- `"terminated"`
- `"stopped"`
- `"connection_lost"`

subnet_id

Die Subnetz-ID der Instance (Zeichenfolge).

virtualization_type

Der Virtualisierungstyp der Instance (Zeichenfolge).

Data Bag für Layer (aws_opsworks_layer)

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu

migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Enthält die Einstellungen für einen Layer.

Das folgende Beispiel zeigt, wie Sie mit der Chef-Suchfunktion ein einzelnes Data Bag-Element (und anschließend mehrere Data Bag-Elemente) durchsuchen und Nachrichten mit den Namen und den Kurznamen der Layer ins Chef-Protokoll schreiben:

```
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	Name
Pakete	shortname	Typ
volume_configurations		

ecs_cluster_arn

Wenn dem Layer ein Amazon ECS-Cluster zugewiesen ist, der Amazon-Ressourcenname (ARN) (Zeichenfolge) des Amazon ECS-Clusters.

verschlüsselt

`true`, wenn das EBS-Volume verschlüsselt ist, andernfalls `false` (Boolesch).

layer_id

Die Layer-ID, bei der es sich um eine GUID handelt, die von AWS OpsWorks Stacks generiert wird und die Ebene eindeutig identifiziert (Zeichenfolge).

Name

Der Layer-Name, mit dem der Layer in der Konsole angezeigt wird (Zeichenfolge). Der Name kann benutzerdefiniert sein, Eindeutigkeit ist nicht erforderlich.

Pakete

Eine Liste der zu installierenden Pakete (Liste aus Zeichenfolgen).

shortname

Der benutzerdefinierte Kurzname des Layers (Zeichenfolge).

Typ

Der Layer-Typ, der bei Chef 12 Linux und Chef 12.2 Windows immer auf `"custom"` festgelegt ist (Zeichenfolge).

volume_configurations

Eine Liste der Amazon EBS-Volume-Konfigurationen.

iops

Die Anzahl der E/A-Vorgänge pro Sekunde, die das Volume unterstützt.

mount_point

Das Verzeichnis für den Mounting-Punkt des Volumes.

number_of_disks

Gibt die Anzahl von Datenträgern im Volume an.

raid_level

Die RAID-Konfiguration des Volumes.

size

Die Volume-Größe in GiB.

volume_type

Der Volume-Typ: Allzweck, Magnetic, bereitgestellte IOPS, Throughput Optimized HDD oder Cold HDD.

Amazon RDS-Datentasche (aws_opsworks_rds_db_instance)

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Ein Satz von Datensackinhalten, der die Konfiguration einer Amazon Relational Database Service (Amazon RDS) -Instance wie folgt spezifiziert:

address	db_instance_identifier	db_password
db_user	engine	rds_db_instance_arn
Region		

Das folgende Beispiel zeigt, wie Sie die Chef-Suche verwenden, um ein einzelnes Datenbeutelelement und dann mehrere Datenbeutelelemente zu durchsuchen, um Nachrichten mit den Adressen und Datenbankmodultypen der Amazon RDS-Instances in das Chef-Protokoll zu schreiben:

```
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

Der DNS-Name der Instance.

port

Der Instance-Port.

db_instance_identifier

Die Instance-ID.

db_password

Das Instance-Masterpasswort.

db_user

Der Instance-Masterbenutzername.

engine

Die Datenbank-Engine der Instance, z. B. `mysql`.

rds_db_instance_arn

Die Amazon-Ressourcenname (ARN) der Instance.

Region

Die AWS-Region der Instance, z. B. `us-west-2`.

Data Bag für Stacks (aws_opsworks_stack)

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Enthält die Einstellungen für einen Stack.

Das folgende Beispiel zeigt, wie Sie mithilfe der Chef-Suchfunktion Nachrichten mit dem Stack-Namen und der Quell-URL des Rezeptbuchs ins Chef-Protokoll schreiben:

```
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	Name
Region	stack_id	use_custom_cookbooks
vpc_id		

arn

Der Amazon-Ressourcenname (ARN) des Stacks (Zeichenfolge).

custom_cookbooks_source

Diese Inhalte geben das Quell-Repository des benutzerdefinierten Rezeptbuchs an.

Typ

Der Repository-Typ (Zeichenfolge). Gültige Werte sind:

- `"archive"`
- `"git"`
- `"s3"`

URL

Die Repository-URL, z. B. `"git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git"` (Zeichenfolge)

username

Der Benutzername für private Repositorys und `null` für öffentliche Repositorys (Zeichenfolge). Bei privaten Amazon Simple Storage Service (Amazon S3) -Buckets ist der Inhalt auf den Zugriffsschlüssel festgelegt.

password

Das Passwort für private Repositorys und `null` für öffentliche Repositorys (Zeichenfolge). Bei privaten S3-Buckets sind diese Inhalte auf den geheimen Schlüssel festgelegt.

ssh_key

Ein SSH-Bereitstellungsschlüssel für den Zugriff auf private Git-Repositorys und `null` für öffentliche Repositorys (Zeichenfolge).

Änderung

Falls das Repository über mehrere Branches verfügt, geben die Inhalte den Branch oder die Version der App an, z. B. `"version1"` (Zeichenfolge). Andernfalls lautet der Wert `null`.

Name

Der Stack-Name (Zeichenfolge).

Region

Die AWS-Region des Stacks (Zeichenfolge).

stack_id

Diese GUID identifiziert den Stack (Zeichenfolge).

use_custom_cookbooks

Gibt an, ob benutzerdefinierte Rezeptbücher aktiviert sind (Boolescher Wert).

vpc_id

Sofern der Stack in einer VPC ausgeführt wird, wird hier die entsprechende VPC-ID angegeben (Zeichenfolge).

Data Bag für Benutzer (aws_opsworks_user)

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Nutzungsdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Enthält die Einstellungen für einen Benutzer.

Das folgende Beispiel zeigt, wie Sie mit der Chef-Suchfunktion ein einzelnes Data Bag-Element (und anschließend mehrere Data Bag-Elemente) durchsuchen und Nachrichten mit den Benutzernamen und den Amazon-Ressourcennamen (ARNs) der Benutzer ins Chef-Protokoll schreiben:

```
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

Gibt an, ob der Benutzer über Administratorrechte verfügt (Boolescher Wert).

iam_user_arn

Der Amazon-Ressourcenname (ARN) des Benutzers (Zeichenfolge).

remote_access

Gibt an, ob sich der Benutzer mithilfe von RDP an der Instance anmelden kann (Boolescher Wert).

ssh_public_key

Der öffentliche Schlüssel des Benutzers, wie er über die AWS OpsWorks Stacks-Konsole oder API bereitgestellt wird (Zeichenfolge).

unix_user_id

Die Unix-ID des Benutzers (Ziffer).

username

Der Benutzername (Zeichenfolge).

OpsWorks Agentenänderungen

> **Important**
>
> Der AWS OpsWorks Stacks Dienst hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Versionen des Chef 12-Agenten

In der folgenden Tabelle werden wichtige Änderungen am Chef 12-Agenten beschrieben, den AWS OpsWorks Stacks auf den von ihm verwalteten Instances installiert.

Agent-Version	Beschreibung	Veröffentlichungsdatum
4042	• Diese Agentenversion enthält nur geringfügige Änderungen ohne neue Funktionen	07. Februar 2023
4041	• Diese Agentenversion enthält nur geringfügige Änderungen ohne neue Funktionen • Amazon CA-Zertifikate aktualisieren	27. Januar 2023
4040	• Diese Agentenversion enthält nur geringfügige Änderungen ohne neue Funktionen	22. Juli 2022
4039	• Korrigiert die ECS-Integration für Ubuntu-AMIs	30. April 2020
4038	• Fehler beim Senden von Instance-Statistiken während der Umschaltung Winter-/Sommerzeit behoben	5. März 2020

Agent-Version	Beschreibung	Veröffentlichungsdatum
	• `no_proxy`-Umgebungsvariable wird beim Herunterladen und Installieren des Agenten berücksichtigt	
4037	• Unterstützung für das Signieren von Anforderungen an S3-URLs ohne Region mithilfe von Sigv4 hinzugefügt • Unterstützung für das Signieren von S3-Anforderungen mit SigV2 entfernt	4. Juni 2019
4035	• Fehler bei ECS-Setup beheben • Doppelte `fstab`-Einträge nach einem Instance-Typwechsel beheben	8. Mai 2019
4033	• Unterstützung für Ubuntu 18.04 hinzugefügt • Fehler bei der Installation in Amazon Linux 2 beheben	26. November 2018
4032	• Support für Amazon Linux 2 hinzugefügt	24. Oktober 2018
4031	• Support für Amazon Linux 2018.03 hinzugefügt • Support öffentlicher S3-Archive, die in einem anderen Konto gehostet werden	15. August 2018
4030	• Volume-Handling für c5d-Instances behoben	31. Mai 2018
4029	• Installation von `nvme-cli` auf Ubuntu 14.04 • Volume-Mounting auf c5-, m5-Instances behoben • Hostnamen beim Neustart immer beibehalten	2. Mai 2018
4028	• `monit`-Konfiguration für CentOS behoben	20. März 2018

Agent-Version	Beschreibung	Veröffentlichungsdatum
4027	• Support für das Mounten von NVMe-Volumes auf Ubuntu 14.04 (`nvme-cli` muss manuell installiert werden) • Die `name`-Eigenschaft ist für Volumes nicht erforderlich	17. Februar 2018
4026	• Mounten NVMe-basierter EBS-Volumes mit EBS-Volume-ID • EBS-Volume-Mounting auf i3-Instances behoben • Reihenfolge gemounteter EBS-Volumes auf c5-, m5-Instances behoben	31. Januar 2018
4025	• Korrektur für die Verarbeitung von NVMe-Geräten	13. Dezember 2017
4024	• Unterstützung von Amazon Linux 2017.09 hinzugefügt	5. Dezember 2017
4023	• Unterstützung für die CloudWatch Logs-Integration hinzufügen	2. April 2017
4022	• Die Chef Client-Version auf 12.18.31 aktualisiert	1. Februar 2017
4021	• Die Proxyverarbeitung verbessert	16. Dezember 2016
4020	• Die Chef Client-Version auf 12.16.42 aktualisiert	8. Dezember 2016

Agent-Version	Beschreibung	Veröffentlichungsdatum
4019	• Quell-Proxy-Variablen während der Agenteninstallation • Red Hat Enterprise Linux 7 verwendet nun `systemd` anstelle von `monit` • Richten Sie EPEL nicht auf Red Hat Enterprise Linux 7 ein • Verwenden Sie `flock` anstelle von `lockrun.c` für die Verarbeitung von Sperren • Vermeiden Sie ungerade Ausgaben von `ps -p1`, wenn Sie `systemd` überprüfen	19. Oktober 2016
4018	• Die Chef Client-Version auf 12.13.37 aktualisiert • Support für Amazon Linux 2016.09 hinzugefügt	25. August 2016
4017	• Die Chef Client-Version auf 12.12.15 aktualisiert	10. August 2016
4016	• Deinstallationsproblem des Agenten auf Systemen, in denen `monit` nicht verwendet wird, behoben	23. Juni 2016
4015	• ECS-Setup für Amazon Linux 2016.03 korrigiert	17. Juni 2016
4011	• Die Chef Client-Version auf 12.10.24 aktualisiert • Verarbeitung des Protokolluploads verbessert	19. Mai 2016
4008	• Support für Amazon Linux 2016.03 hinzugefügt • Zeitbeschränkung hinzugefügt, um die Installation zu bündeln • xfs zu /etc/filesystems hinzugefügt, falls vorhanden	16. März 2016

Agent-Version	Beschreibung	Veröffentlichungsdatum
4007	• Die Chef Client-Version auf 12.7.2 aktualisiert • Verbesserungen für die Fehlerbehandlung für lokale Instances (gehostete Server außerhalb von AWS) • Verbesserung der Kompatibilität mit dem neuesten Chef-Sugar • Wiederholen des Archiv-Downloads für die Bereitstellung	4. März 2016
4006	• Die Chef Client-Version auf 12.6.0 aktualisiert • Installieren Sie die Pakete "libxml2-devel/libxml2-dev" und "libxslt-devel/libxslt-dev" nicht während der Installation des Agenten	21. Januar 2016
4005	• ec2-Import korrigiert, indem ec2-Daten in Ohai für die ec2-Infrastruktur immer aktiviert werden	17. Dezember 2015
4004	• AWS OpsWorks Stacks-Unterstützung für Chef 12 Linux-Chef Client 12.5.1	3. Dezember 2015

Versionen des Chef 11.10-Agenten

In der folgenden Tabelle werden wichtige Änderungen am Chef 11.10-Agenten beschrieben, den AWS OpsWorks Stacks auf den von ihm verwalteten Instances installiert.

Agent-Version	Beschreibung	Veröffentlichungsdatum
3456	• Diese Agent-Version enthält nur geringfügige Änderungen ohne neue Funktionen • Amazon CA-Zertifikate aktualisieren	27. Januar 2023
3455	• Diese Agentenversion enthält nur geringfügige Änderungen ohne neue Funktionen	1. November 2022

Agent-Version	Beschreibung	Veröffentlichungsdatum
3454	• Korrigiert die ECS-Integration für Ubuntu-AMIs	28. April 2020
3453	• Fehler beim Senden von Instance-Statistiken während der Umschaltung Winter-/Sommerzeit behoben • Fehler durch fehlende Pakete im RHEL7-Setup behoben • `no_proxy`-Umgebungsvariable wird beim Herunterladen und Installieren des Agenten berücksichtigt	5. März 2020
3452	• Keine Region in die Amazon S3-URL für den virtuellen Pfad aufnehmen, wenn sie in `us-east-1` ist • Extrahieren und Hochladen interner Rezeptbücher in Stage-Regions-spezifische Buckets • `fstab`-Einträge für Chef 11.10 behoben • SigV2-Nutzung für S3 entfernen und die Region für den Bucket in der Anforderung abrufen	13. August 2019
3451	• Unterstützung für Ruby 2.6.1 hinzugefügt	20. März 2019
3450	• Standard-EBS-Attribute beheben • CloudWatchLogs Agenteninstallation für Amazon Linux 2 korrigieren • Bundler-Installation für rubygem Versionen ab 2.6.14 beheben • Unterstützung für öffentliche S3-Archive beheben	3. Dezember 2018
3449	• Volume-Handling für c5d-Instances behoben • RAID-Array-Support auf NVMe-Gerät-Instances behoben	5. Juni 2018

Agent-Version	Beschreibung	Veröffentlichungsdatum
3448	• Upgrade auf die Standard-Version 2.3 von Ruby 2.3.7 • Mounten von EBS-Volumes auf NVMe-basierten Instances auf Ubuntu 14.04-Instances behoben • Support öffentliche Amazon S3 S3-Archive, die auf einem anderen Konto gehostet werden • `opsworks-agent` Boot-Probleme auf Red Hat Enterprise Linux-Instances behoben	8. Mai 2018
3447	• Mounten NVMe-basierter EBS-Volumes mit EBS-Volume-ID • EBS-Volume-Mounting auf i3-Instances behoben • Reihenfolge gemounteter EBS-Volumes auf c5, m5 behoben • Update auf die Standard-Version 2.3 von Ruby 2.3.6	31. Januar 2018
3446	• Korrektur für die Verarbeitung von NVMe-Geräten • Update auf die Standard-Version 2.3 von Ruby 2.3.5	14. Dezember 2017
3445	• Unterstützung von Amazon Linux 2017.09 hinzugefügt • Update auf die Standard-Version 2.2 von Ruby 2.2.8	31. Oktober 2017
3444	• Unterstützung für CloudWatch Logs hinzufügen	1. April 2017
3443	• Die Proxyverarbeitung verbessert	15. Dezember 2016

Agent-Version	Beschreibung	Veröffentlichungsdatum
3442	• Update auf die Standard-Version 2.3 von Ruby 2.3.3 • Update auf die Standard-Version 2.2 von Ruby 2.2.6	6. Dezember 2016
3441	• Quell-Proxy-Variablen während der Agenteninstallation	21. Oktober 2016
3440	• Support für Amazon Linux 2016.09 hinzugefügt	13. September 2016
3439	• Kleinere Änderungen; keine neuen Funktionen	29. Juli 2016
3438	• Unterstützung für Ruby 2.3.1 hinzugefügt • Instance-Registrierung mit Anmeldeinformationen vom IAM-Instance-Profil verbessert • `s3curl.pl` -Überreste entfernt • ECS-Setup für Amazon Linux 2016.03 korrigiert	17. Juni 2016
3437	• Update auf die Standard-Version 2.2 von Ruby 2.2.5	4. Mai 2016
3436	• Update EPEL URL für Red Hat Enterprise Linux. WICHTIG: ohne diese Änderung können Red Hat Enterprise Linux-Instances nicht starten.	18. April 2016
3435	• Update auf die Standard-Version 2.1 von Ruby 2.1.9 • Verbessern Sie die Handhabung von Amazon S3- und Archivbereitstellungen	6. April 2016
3434	• Support für Amazon Linux 2016.03 hinzugefügt • Wiederholung von Paketinstallationen	16. März 2016

Agent-Version	Beschreibung	Veröffentlichungsdatum
3433	• Einige Verbesserungen für lokale Instances (Server, die außerhalb von gehostet werden) AWS • Verbesserung der Kompatibilität mit dem neuesten `chef-sugar` • Wiederholen des Archiv-Downloads für die Bereitstellung • Ruby Gems Installations-URL korrigiert	27. Februar 2016
3432	• Handling von Sonderzeichen in Bucket-Namen verbessert • Update von `s3_file` auf Version 2.6.6 • Überspringen des Mountens von Volumes ohne angegebenen Mounting-Punkt • `unicorn` immer neu starten, statt es anzuhalten und wieder zu starten, um Ausfallzeiten während der Bereitstellung zu vermeiden • Benutzerdefiniertes Rezeptbuch für den `setup`-Befehl immer aktualisieren • Nach dem Erstellen von RAID-Arrays `initramfs` aktualisieren, um Probleme bei der Gerätezweisung beim Neustart zu vermeiden	20. Januar 2016
3431	• Problem bei der Installation der Gems `passenger` und `unicorn` in der Rails-Ebene behoben • Aktualisieren der Standardversionen 2.0, 2.1 und 2.2 von Ruby auf 2.0.0p648, 2.1.8 und 2.2.4 • Zulassen, dass `postgres`-Paketnamen in einer benutzerdefinierten JSON-Datei festgelegt werden • Update der Node.js-Standardversion auf 0.12.9	22. Dezember 2015
3430	• Kleinere Änderungen; keine neuen Funktionen	25. November 2015

Agent-Version	Beschreibung	Veröffentlichungsdatum
3429	• Verbessern Sie OpsWorks Agent Daemonize (schließen Sie stdout/stderr) • Stabilität der `s3_file`-Ressource verbessert (Wiederholungen, Auffangen von Ausnahmen)	18. November 2015
3428	• `postgres`-Adaptererkennung basierend auf der Gemfile hinzugefügt, korrigiert https://github.com/aws/opsworks-cookbooks/issues/136	17. Juni 2016
3427	• Ein Problem beim Abrufen von Anmeldeinformationen in den Agenten behoben • Aktualisieren der Standardversionen 2.0, 2.1 und 2.2 von Ruby auf 2.0.0p647, 2.1.7 und 2.2.3	11. September 2015
3426	• `aws-sdk` auf 1.65.0 aktualisiert • Verbesserung des Downloads von Amazon S3 durch `s3curl` Ersetzen durch `s3_file cookbook` • Änderung der Node.js-Standardversion auf 0.12.7 • Protokollierung für Node.js-Apps hinzugefügt. STDERR und STDOUT im Verzeichnis „`shared/log` “ protokolliert und rotiert • Update des Checkouts von Rezeptbuch-Untermodulen explizit gemacht • Problemumgehung für https://github.com/aws/opsworks-cookbooks/issues/213 hinzugefügt, der überprüft, ob Bind-Mounts erfolgt sind, bevor das Verzeichnis `deploy` erstellt wird	27. August 2015
3425	• ECS-Support für Amazon Linux und Ubuntu	27. Juli 2015
3424	• Kleinere Änderungen; keine neuen Funktionen	9. Juli 2015

Agent-Version	Beschreibung	Veröffentlichungsdatum
3422	• Vollständiger Support von Red Hat Enterprise Linux 7 • Die `/etc/hosts` -Generation widerstandsfähiger gegenüber Fehlern gemacht	29. Juni 2015
3421	• Option zum Überschreiben des Datenbank paketnamens für Red Hat Enterprise Linux 7 • Die `monit systemd`-Konfigurationsdatei aktualisiert, um zu verhindern, dass `systemd` das `kill`-Signal an Prozesse sendet, die von `monit` überwacht werden	11. Juni 2015

AWS OpsWorks Stacks-Ressourcen

> **Important**
>
> Der AWS OpsWorks Stacks Service hat am 26. Mai 2024 das Ende seiner Lebensdauer erreicht und wurde sowohl für neue als auch für bestehende Kunden deaktiviert. Wir empfehlen Kunden dringend, ihre Workloads so bald wie möglich auf andere Lösungen zu migrieren. Wenn Sie Fragen zur Migration haben, wenden Sie sich an das AWS Support Team auf AWS re:POST oder über den AWS Premium-Support.

Die folgenden verwandten Ressourcen bieten Ihnen nützliche Informationen für die Arbeit mit diesem Service.

Referenz-Handbücher, Tools und Support-Ressourcen

Mehrere hilfreiche Handbücher, Foren, Kontaktinformationen und andere Ressourcen sind von AWS OpsWorks Stacks und Amazon Web Services erhältlich.

- AWS OpsWorks Stacks-API-Referenz — Beschreibungen, Syntax und Anwendungsbeispiele zu AWS OpsWorks Stacks-Aktionen und Datentypen, einschließlich allgemeiner Parameter und Fehlercodes.
- AWS OpsWorks Häufig gestellte technische Fragen zu Stacks — Die häufigsten Fragen, die Entwickler zu diesem Produkt gestellt haben.
- AWS OpsWorks Stacks-Versionshinweise — Ein allgemeiner Überblick über die aktuelle Version. Dieses Dokument führt speziell alle neuen Funktionen, Korrekturen und bekannten Probleme auf.
- AWS-Tools für PowerShell — Eine Reihe von PowerShell Windows-Cmdlets, die die Funktionalität von AWS SDK for .NET in der PowerShell Umgebung verfügbar machen.
- AWS-Befehlszeilenschnittstelle — Eine einheitliche Befehlszeilensyntax für den Zugriff auf AWS-Services. Die AWS-CLI verwendet einen einzelne Einrichtungsprozess, um den Zugriff für alle unterstützten Services zu aktivieren.
- AWS OpsWorks Stacks-Befehlszeilenreferenz — AWS OpsWorks Stacks-spezifische Befehle zur Verwendung an einer Befehlszeilenaufforderung.

- Kurse und Workshops — Links zu rollen- und Spezialkursen sowie zu Übungen zum Selbststudium, mit denen Sie Ihre Fähigkeiten verbessern und praktische Erfahrungen sammeln können. AWS
- AWS Developer Center — Erkunden Sie Tutorials, laden Sie Tools herunter und erfahren Sie mehr über Veranstaltungen für Entwickler. AWS
- AWS Entwicklertools — Links zu Entwicklertools, SDKs, IDE-Toolkits und Befehlszeilentools für die Entwicklung und Verwaltung von AWS Anwendungen.
- Ressourcencenter für die ersten Schritte — Erfahren Sie, wie Sie Ihre AWS-Konto Anwendung einrichten, der AWS Community beitreten und Ihre erste Anwendung starten.
- Praktische Tutorials — Folgen Sie den step-by-step Tutorials, um Ihre erste Anwendung zu starten. AWS
- AWS Whitepapers — Links zu einer umfassenden Liste von technischen AWS Whitepapers zu Themen wie Architektur, Sicherheit und Wirtschaft, die von Solutions Architects oder anderen technischen Experten verfasst wurden. AWS
- AWS Support Center — Die zentrale Anlaufstelle für die Erstellung und Verwaltung Ihrer Fälle. AWS Support Enthält auch Links zu anderen hilfreichen Ressourcen wie Foren, häufig gestellten Fragen zu technischen Fragen, dem Status des Dienstes und AWS Trusted Advisor.
- AWS Support— Die wichtigste Webseite mit Informationen über AWS Support einen Support-Kanal mit schnellen Reaktionszeiten one-on-one, der Sie bei der Entwicklung und Ausführung von Anwendungen in der Cloud unterstützt.
- Kontakt – Zentraler Kontaktpunkt für Fragen zu AWS -Abrechnung, Konten, Ereignissen Missbrauch und anderen Problemen.
- AWS Nutzungsbedingungen der Website — Detaillierte Informationen zu unseren Urheberrechten und Marken, zu Ihrem Konto, Ihrer Lizenz und Ihrem Zugriff auf die Website sowie zu anderen Themen.

AWS Kits für die Softwareentwicklung

Amazon Web Services bietet Softwareentwicklungskits für den Zugriff auf AWS OpsWorks Stacks aus verschiedenen Programmiersprachen. Die SDK-Bibliotheken automatisieren eine Reihe von allgemeinen Aufgaben, wie z. B. kryptografisches Signieren Ihrer Serviceanfragen, Wiederholen von Anfragen oder die Verarbeitung von Fehlermeldungen.

- AWS SDK for Java— Einrichtung und andere Dokumentation

- AWS SDK for .NET— Einrichtung und andere Dokumentation.
- AWS SDK for PHP — Dokumentation
- AWS SDK for Ruby— Dokumentation
- andere Unterlagen
- AWS SDK for Python (Boto)— Einrichtung und andere Dokumentation

Open-Source-Software

AWS OpsWorks Stacks umfasst eine Vielzahl von Open-Source-Softwarepaketen, die ihren jeweiligen Lizenzen unterliegen. Weitere Informationen finden Sie hier:

- Öffnen Sie bei Chef 12 Linux-Instances die Datei `THIRD_PARTY_LICENSES` im Verzeichnis `/opt/aws/opsworks/current` auf der Instance.
- Laden Sie für Chef 11.10 und frühere Versionen für Linux das Dokument OpsWorks Linux Agent Attributions als PDF herunter.

AWS OpsWorks Historie des Dokumentes

Änderung	Beschreibung	Datum
Aktualisierungen für Stacks AWS OpsWorks	Sie können jetzt das Tool Detach in Place verwenden, um Ihre OpsWorks Instances vom OpsWorks Stacks-Service zu trennen. Weitere Informationen finden Sie unter Verwenden des Tools AWS OpsWorks Stacks Detach in Place in diesem Handbuch.	11. April 2024
Aktualisierungen für AWS OpsWorks Stacks	Sie können Ihre AWS OpsWorks Stacks jetzt mithilfe eines Migrationsskripts zu AWS Systems Manager Application Manager migrieren. Weitere Informationen finden Sie in diesem Handbuch unter Migrieren Ihrer AWS OpsWorks Stacks Anwendungen zu AWS Systems Manager Application Manager.	22. Dezember 2022
Aktualisierungen für und AWS OpsWorks for Chef AutomateAWS OpsWorks for Puppet Enterprise	Es ist jetzt ein Verfahren zur Problembehandlung verfügbar, das beschreibt, was Sie tun können, wenn die Systemwartung für Ihren AWS OpsWorks for Chef Automate oder OpsWorks für den Puppet Enterprise-Server fehlschlägt. Weitere Informationen finden Sie unter Die Systemwar	29. September 2022

	tung schlägt für den Chef Automate-Server fehl oder Die Systemwartung schlägt für den Puppet Enterprise-Server fehl. in diesem Handbuch.	
Aktualisierungen für AWS OpsWorks for Chef Automate und AWS OpsWorks for Puppet Enterprise	Ein Verfahren zur Fehlerbehebung ist jetzt verfügbar, wenn Ihr AWS OpsWorks for Chef Automate oder OpsWorks für den Puppet Enterprise-Server in einen `Connection lost` Zustand übergeht. Weitere Informationen finden Sie in diesem Handbuch unter Chef Automate-Server befindet sich in einem `Connection lost` Zustand oder Puppet Enterprise-Server befindet sich in einem `Connection lost` Zustand.	23. März 2022

Aktualisierungen für Stacks AWS OpsWorks	Als bewährte Sicherheitsmethode können Sie jetzt einen `aws:SourceArn` oder einen `aws:SourceAccount` Bedingungsschlüssel (oder beides) hinzufügen, um Richtlinien für Vertrauensbeziehungen zu erstellen, die es AWS OpsWorks Stacks ermöglichen, Aufgaben in anderen AWS Diensten auszuführen. Weitere Informationen finden Sie in diesem Leitfaden unter Dienstübergreifendes Vermeiden verwirrter Stellvertreter in AWS OpsWorks Stacks.	4. März 2022

Aktualisierungen für und AWS OpsWorks for Chef AutomateAWS OpsWorks for Puppet Enterprise	Als bewährte Sicherheitsmethode können Sie jetzt einen `aws:SourceArn` oder einen `aws:SourceAccount` Bedingungsschlüssel (oder beides) hinzufügen, um Richtlinien OpsWorks für Vertrauensbeziehungen hinzuzufügen, die Puppet Enterprise den Zugriff auf die Ausführung von Aufgaben in anderen AWS Diensten ermöglichen AWS OpsWorks for Chef Automate und ermöglichen. Weitere Informationen finden Sie unter Dienstübergreifendes Vermeiden verwirrter Stellvertreter in diesem Handbuch.	10. Januar 2022
Aktualisierungen für und AWS OpsWorks for Chef AutomateAWS OpsWorks for Puppet Enterprise	AWS OpsWorks for Chef Automate und OpsWorks für Puppet Enterprise haben die verwalteten Richtlinien `AWSOpsWorksCMServiceRole` aktualisiert und speichern nun Geheimnisse in AWS Secrets Manager. `AWSOpsWorksCMInstanceProfileRole`	3. Mai 2021

Aktualisierungen für AWS OpsWorks for Puppet Enterprise	Die Engine-Version eines Servers OpsWorks für Puppet Enterprise, den Sie in der Konsole erstellen, ist jetzt 2019.8.5. Mithilfe der API können Sie entweder die Version `2019` oder `2017` bei der Erstellung eines Puppet Enterprise-Servers angeben. Die `DescribeServers` API gibt jetzt ein Attribut zurück, das `PUPPET_API_CRL` in ihren Ergebnissen aufgerufe n wird. Dieses Attribut enthält eine Zertifikatssperrliste für den internen Gebrauch.	28. April 2021
AWS OpsWorks Stacks verwendet eine neue verwaltet e Richtlinie	AWS OpsWorks Stacks hat die verwaltete Richtlinie geändert, die Berechtigungen zur Ausführung aller Aktionen in AWS OpsWorks Stacks beinhaltet. Die neue Richtlinie lautet. AWSOpsWorks_FullAc cess Weitere Informationen zu den Berechtigungen in dieser Richtlinie finden Sie unter Beispielrichtlinien.	19. Februar 2021
Migrieren Sie AWS OpsWorks Stacks Stacks von EC2-Class ic zu einer VPC	Es wurde eine Dokumenta tion hinzugefügt, die beschreib t, wie ein AWS OpsWorks Stacks Stack von EC2-Classic zu einer VPC migriert wird.	29. September 2020

Regenerieren Sie ein Starterkit für und AWS OpsWorks for Chef AutomateAWS OpsWorks for Puppet Enterprise	Es wurde eine Dokumentation hinzugefügt, die beschreibt, wie das Starterkit für einen AWS OpsWorks for Chef Automate oder einen AWS OpsWorks for Puppet Enterprise Server regeneriert wird.	29. Juli 2020
AWS OpsWorks for Puppet Enterprise ermöglicht es Ihnen, einen Server zu erstellen, der eine benutzerdefinierte Domäne, ein benutzerdefiniertes Zertifikat und einen privaten Schlüssel verwendet	Sie können jetzt einen Server OpsWorks für Puppet Enterprise erstellen, der eine benutzerdefinierte Domäne, ein Zertifikat und einen privaten Schlüssel verwendet. Sie können einen vorhandenen Puppet Enterprise-Server so aktualisieren, dass er eine benutzerdefinierte Domäne verwendet, indem Sie einen Server anhand einer Sicherung eines vorhandenen Servers erstellen.	17. April 2020

AWS OpsWorks for Chef Automate und unterstützt AWS OpsWorks for Puppet Enterprise jetzt Tagging in der Konsole	Sie können jetzt Tags zu einem AWS OpsWorks for Chef Automate Server oder einem AWS OpsWorks for Puppet Enterprise Master oder zu Server-Backups hinzufüge n, indem Sie entweder den AWS Management Console oder den AWS CLI verwenden . Weitere Informationen finden Sie unter Mit Tags arbeiten (Chef) oder Mit Tags arbeiten (Puppet).	26. Februar 2020
AWS OpsWorks for Chef Automate vereinfacht das Upgrade vorhandener Chef Automate 1-Server auf Chef Automate 2	Sie können berechtigte AWS OpsWorks for Chef Automate Server, auf denen Chef Automate 1 ausgeführt wird, auf Chef Automate 2 aktualisi eren, indem Sie auf der Detailseite Ihres Servers in der Konsole die Option Upgrade starten auswählen oder die `StartMaintenance` API-Aktion ausführen. Weitere Informationen finden Sie unter Upgrade eines AWS OpsWorks for Chef Automate Servers auf Chef Automate 2.	24. Januar 2020

AWS OpsWorks for Chef Automate und AWS OpsWorks for Puppet Enterprise	Dem Leitfaden wurde ein neues Kapitel über Sicherheit in AWS OpsWorks CM (AWS OpsWorks for Chef Automate und AWS OpsWorks for Puppet Enterprise) hinzugefügt.	23. Dezember 2019
AWS OpsWorks for Chef Automate und AWS OpsWorks for Puppet Enterprise unterstützt Tagging	Sie können jetzt Tags zu einem AWS OpsWorks for Chef Automate Server oder einem AWS OpsWorks for Puppet Enterprise Master oder zu Server-Backups hinzufügen, indem Sie den AWS CLI verwenden. AWS OpsWorks CM unterstützt jetzt die Tag-basierte Autorisierung.	18. Dezember 2019
AWS OpsWorks for Chef Automate ermöglicht es Ihnen, einen Server zu erstellen, der eine benutzerdefinierte Domäne, ein Zertifikat und einen privaten Schlüssel verwendet	Sie können jetzt einen AWS OpsWorks for Chef Automate 2.0-Server erstellen, der eine benutzerdefinierte Domäne, ein Zertifikat und einen privaten Schlüssel verwendet. Sie können einen vorhandenen Chef Automate 2.0-Server so aktualisieren, dass er eine benutzerdefinierte Domäne verwendet, indem Sie einen Server anhand einer Sicherung eines vorhandenen Servers erstellen.	22. Oktober 2019

AWS OpsWorks Stacks unterstützt jetzt Ruby 2.6.1	AWS OpsWorks Stacks unterstützt Ruby 2.6.1 auf Rails App Server-Ebenen in Chef 11.10-Stacks.	2. Mai 2019
AWS OpsWorks for Chef Automate unterstützt jetzt Chef Automate 2.0	Auf neuen AWS OpsWorks for Chef Automate Servern wird Chef Automate 2.0 laufen, das Updates für Chef InSpec, neue Funktionen für Compliance-Scannen und Reporting sowie Chef Infra beinhaltet.	30. April 2019
AWS OpsWorks for Chef Automate und AWS OpsWorks for Puppet Enterprise	Sie können es jetzt verwenden AWS CloudFormation , um einen AWS OpsWorks for Chef Automate Server oder einen AWS OpsWorks for Puppet Enterprise Masterser ver zu erstellen.	24. Januar 2019
AWS OpsWorks Stacks	AWS OpsWorks Stacks unterstützt jetzt Instanzen, auf denen Ubuntu 18.04 LTS in Chef 12-Stacks ausgeführt wird.	18. Dezember 2018
AWS OpsWorks für Puppet Enterprise	Es wurde ein Verfahren zum Einrichten einer SSH-basierten Verbindung zu einem Kontroll-Repository hinzugefügt, das verwendet. CodeCommit	3. Dezember 2018
AWS OpsWorks Stacks	AWS OpsWorks Stacks unterstützt jetzt Instances, auf denen Amazon Linux 2 in Chef 12-Stacks ausgeführt wird.	15. November 2018

AWS OpsWorks Stacks	AWS OpsWorks Stacks unterstützt jetzt Instances , auf denen Amazon Linux 2018.03 in Chef 11.10-Stacks ausgeführt wird.	23. Oktober 2018
AWS OpsWorks Stacks	AWS OpsWorks Stacks unterstützt jetzt Instances, auf denen Amazon Linux 2018.03 in Chef 12-Stacks ausgeführt wird.	23. August 2018
AWS OpsWorks for Chef Automate und für Puppet Enterprise OpsWorks	OpsWorks für Puppet Enterprise wurde auf PE 2018.1.2 aktualisiert. AWS OpsWorks for Chef Automate wurde auf Chef Automate 1.8.68 aktualisiert.	29. Juni 2018

- AWS OpsWorks for Chef Automate und OpsWorks für Puppet Enterprise API-Version: 2016-11-01
- AWS OpsWorks Stacks API-Version: 2016-03-08
- Letzte Aktualisierung der Dokumentation: 2024-04-11

Frühere Aktualisierungen

In der folgenden Tabelle sind wichtige Änderungen in jeder Version des AWS OpsWorks - Benutzerhandbuchs vor Juni 2018 beschrieben.

Beschreibung	Datum
AWS OpsWorks Die Stacks Chef-Version für Windows-basierte Stacks wurde auf 12.22 aktualisiert; die Ruby-Version ist jetzt 2.3.6.	19. April 2018

Beschreibung	Datum
Neue Verfahren zum Erstellen eines AWS OpsWorks for Chef Automate Servers oder eines Masters OpsWorks für Puppet Enterpris e mithilfe von. AWS CLI	23. März 2018
Die Version von Chef Automate wurde auf 1.8 aktualisiert; die Einrichtung von Chef Compliance wurde durch das Hinzufügen des `opsworks-audit` Kochbuches vereinfacht.	5. März 2018
Unterstützung für AWS OpsWorks Stacks-Ereignisse in Amazon CloudWatch Events hinzugefügt.	20. Februar 2018
Unterstützung für neue EBS-Volumetypen in AWS OpsWorks Stacks und eine neue API hinzugefügt. DescribeOperatingSystems	25. Januar 2018
OpsWorks für Puppet Enterprise und unterstützt AWS OpsWorks for Chef Automate jetzt die Auswahl mehrerer Sicherheitsgruppen, wenn Sie einen Server erstellen.	18. Januar 2018
Unterstützung für AWS OpsWorks Stacks in der Region Europa (Paris) hinzugefügt.	19. Dezember 2017
Unterstützung für AWS OpsWorks for Chef Automate und OpsWorks für Puppet Enterprise in sechs weiteren Regionen hinzugefügt und Verfahren zum Erstellen von Backups von AWS OpsWorks for Chef Automate und OpsWorks für Puppet Enterprise-Server in der hinzugefügt. AWS Management Console	18. Dezember 2017
Der neue Service und die Dokumentation OpsWorks für Puppet Enterprise wurden hinzugefügt.	16. November 2017
Unterstützung für Amazon Linux 2017.09 wurde zu AWS OpsWorks Stacks hinzugefügt.	7. November 2017
Unterstützung für Chef Compliance wurde hinzugefügt. AWS OpsWorks for Chef Automate	25. Oktober 2017

Beschreibung	Datum
Unterstützung für Amazon Linux 2017.09 wurde hinzugefügt. AWS OpsWorks for Chef Automate	9. Oktober 2017
Das Thema Systemwartung wurde dem Kapitel hinzugefügt. AWS OpsWorks for Chef Automate	28. Juli 2017
Unterstützung für Tags in AWS OpsWorks Stacks hinzugefügt.	6. Juni 2017
Integration mit CloudWatch Logs hinzugefügt.	10. April 2017
Der neue AWS OpsWorks for Chef Automate Service und die neue Dokumentation wurden hinzugefügt.	1. Dezember 2016
Unterstützung für den regionalen Endpunkt der Region USA Ost (Ohio) hinzugefügt.	12. Oktober 2016
Stacks und Instances, die das Amazon Linux 2016.09-Betriebssystem ausführen, werden nun unterstützt.	30. September 2016
Unterstützung für die Region Asien-Pazifik (Seoul) und neun weitere regionale Endpunkte hinzugefügt.	15. August 2016
Node.js 0.12.15 und Ruby 2.3 werden nun in integrierten Ebenen unterstützt.	6. Juli 2016
Unterstützung für die Region Asien-Pazifik (Mumbai) hinzugefügt.	28. Juni 2016
Stacks und Instances, die das CentOS 7-Betriebssystem ausführen, werden nun unterstützt.	22. Juni 2016
Eine Beschreibung der Komplettlösung CodePipeline und die AWS OpsWorks Stacks-Integration wurden hinzugefügt.	2. Juni 2016
Stacks und Instances, die das Ubuntu 16.04 LTS-Betriebssystem ausführen, werden nun unterstützt.	1. Juni 2016
Unterstützung für Chef 12 Linux und die dazugehörige Dokumentation wurden hinzugefügt.	3. Dezember 2015

Beschreibung	Datum
Eine schrittweise Anleitung zu Node.js für die ersten Schritte wurde hinzugefügt.	14. Juli 2015
Den Rezeptbüchern 101 wurden zwei neue Rezeptbuchbeispiele hinzugefügt.	14. Juli 2015
Agent-Versionsverwaltung wird nun unterstützt.	23. Juni 2015
Die Verwaltung der Agent-Version wird nun unterstützt.	24. Juni 2015
Benutzerdefinierte Windows-AMIs werden nun unterstützt.	22. Juni 2015
Drei neue Themen zu bewährten Methoden wurden hinzugefügt.	11. Juni 2015
Windows-Stacks werden nun unterstützt.	18. Mai 2015
Ein Kapitel zu bewährten Methoden wurde hinzugefügt.	15. Dezember 2014
Unterstützung für Elastic Load Balancing Connection Draining und benutzerdefinierte Shutdown-Timeouts hinzugefügt.	15. Dezember 2014
Unterstützung für die Registrierung von Instances hinzugefügt, die außerhalb von AWS OpsWorks Stacks erstellt wurden.	9. Dezember 2014
Unterstützung für Amazon SWF hinzugefügt.	4. September 2014
Das Zuordnen von Umgebungsvariablen zu Apps und den erweitertem Rezeptbüchern 101 wird nun unterstützt.	16. Juli 2014
Rezeptbücher 101, ein Einführungs-Tutorial für das Implementieren von Rezeptbüchern, wurde hinzugefügt.	16. Juli 2014
Unterstützung für hinzugefügt CloudTrail.	4. Juni 2014
Unterstützung für Amazon RDS hinzugefügt.	14. Mai 2014
Chef 11.10 und Berkshelf werden nun unterstützt.	27. März 2014
Unterstützung für Amazon EBS PIOPS-Volumes hinzugefügt.	16. Dezember 2013

Beschreibung	Datum
Ressourcenbasierte Berechtigungen wurden hinzugefügt.	5. Dezember 2013
Ressourcenverwaltung wurde hinzugefügt.	7. Oktober 2013
VPCs werden nun unterstützt.	29. August 2013
Benutzerdefinierte AMIs und Chef 11.4 werden nun unterstützt.	24. Juli 2013
Konsolen für mehrere Ebenen pro Instance werden nun unterstützt.	1. Juli 2013
Unterstützung für Amazon EBS-gestützte Instances, Elastic Load Balancing und CloudWatch Amazon-Überwachung hinzugefügt.	14. Mai 2013
Erste Version des AWS OpsWorks Stacks-Benutzerhandbuchs.	18. Februar 2013

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.