Guide de l’utilisateur

AWS OpsWorks



Version de l'API 2013-02-18

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

AWS OpsWorks: Guide de l’utilisateur

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

Les marques et la présentation commerciale d’Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n’est pas d’Amazon, d’une manière susceptible de créer une confusion parmi les clients, ou d’une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d’Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Qu'est-ce que c'est AWS OpsWorks ? 1
AWS OpsWorks Des services 1
AWS OpsWorks pour Puppet Enterprise 4
Support régional OpsWorks pour Puppet Enterprise 5
Fin de vie FAQs 6
Comment les clients existants seront-ils affectés par cette fin de vie ? 6
Qu'arrive-t-il à mes serveurs si je n'agis pas ? 7
Est-ce AWS OpsWorks for Puppet Enterprise accepter de nouveaux clients ? 7
La fin de vie aura-t-elle un impact sur Régions AWS tout le monde en même temps ? 7
Quel est le niveau de support technique disponible AWS OpsWorks for Puppet Enterprise ? 7
Je suis actuellement client de OpsWorks for Puppet Enterprise et je dois lancer un serveur sur un compte qui n'utilisait pas le service auparavant. Suis-je capable de le faire ? 7
Y aura-t-il de nouvelles fonctionnalités pour AWS OpsWorks for Puppet Enterprise ? 8
Démarrage 8
Prérequis 8
Créer un serveur maître Puppet 13
Finaliser la configuration 26
Ajouter des nœuds à gérer 30
Se connecter à la console Puppet Enterprise 33
Facultatif : utilisation CodeCommit 37
Créez un maître de marionnettes dans CloudFormation 44
Prérequis 45
Créer un serveur maître Puppet Enterprise dans AWS CloudFormation 45
Mise à jour d'un serveur pour utiliser un domaine personnalisé 52
Prérequis 53
Limites 53
Mise à jour d'un serveur pour utiliser un domaine personnalisé 54
consultez aussi 58
Utilisation des balises 58
Comment fonctionnent les tags dans AWS OpsWorks for Puppet Enterprise 59
Ajouter et gérer des tags dans OpsWorks Puppet Enterprise (console) 61
Ajouter et gérer des balises dans OpsWorks Puppet Enterprise (CLI) 64
consultez aussi 69

Sauvegarde et restauration de serveurs 70
Sauvegardez et OpsWorks pour Puppet Enterprise Server 70
Restaurer un OpsWorks serveur pour Puppet Enterprise 74
Maintenance du système 76
Configuration de la maintenance du système 77
Démarrage de la maintenance du système à la demande 79
Restauration des configurations personnalisées et des fichiers après la maintenance 80
Ajout automatique de nœuds 80
Étape 1 : créer un rôle IAM à utiliser comme profil d'instance 81
Étape 2 : Création d'instances en utilisant un script d'association sans surveillance 82
Suppression de nœuds 83
consultez aussi 84
Supprimer un maître Puppet 84
Étape 1 : Dissociation de nœuds gérés 85
Étape 2 : Suppression du serveur 85
consultez aussi 86
Migrer un serveur Puppet vers Amazon EC2 86
Étape 1 : contactez Puppet pour acheter une licence 87
Étape 2 : Obtenez des informations sur votre serveur OpsWorks pour Puppet Enterprise 87
Étape 3 : effectuez une sauvegarde de votre serveur OpsWorks pour Puppet Enterprise 88
Étape 4 : Lancer une nouvelle EC2 instance 88
Étape 5 : Installation de Puppet Enterprise sur la nouvelle instance EC2 90
Étape 6 : restauration de la sauvegarde sur la nouvelle EC2 instance 90
Étape 7 : Configuration de votre licence Puppet 91
Étape 8 : migrez vos nœuds 91
Étape 9 : Supprimer votre serveur OpsWorks pour Puppet Enterprise 94
En utilisant AWS CloudTrail 94
OpsWorks pour obtenir des informations sur Puppet Enterprise dans CloudTrail 95
Compréhension OpsWorks des entrées du fichier journal de Puppet Enterprise 96
Résolution des problèmes 98
Conseils généraux de résolution des problèmes 99
Résolution d'erreurs spécifiques 99
Aide supplémentaire et support 104
AWS OpsWorks pour Chef Automate 106
Support régional pour AWS OpsWorks for Chef Automate 109
Fin de vie FAQs 110

Comment les utilisateurs existants seront-ils affectés par cette fin de vie ? 111
Qu'arrive-t-il à mes serveurs si je n'agis pas ? 111
Quelles sont les alternatives vers lesquelles je peux effectuer la transition ? 111
Le service accepte-t-il toujours de nouveaux clients ? 111
La fin de vie affectera-t-elle Régions AWS tout le monde en même temps ? 111
Quel est le niveau de support technique disponible ? 112
Je suis actuellement client de OpsWorks for Chef Automate et je dois lancer un serveur sur un compte qui n'utilisait pas le service auparavant. Suis-je capable de le faire ? 112
Y aura-t-il des nouveautés majeures au cours de l'année prochaine ? 112
Mise à niveau vers Chef Automate 2 112
Conditions préalables à la mise à niveau vers Chef Automate 2 112
À propos du processus de mise à niveau 113
Mise à niveau vers Chef Automate 2 (console) 114
Mise à niveau vers Chef Automate 2 (interface de ligne de commande) 114
Restaurer un AWS OpsWorks for Chef Automate serveur vers Chef Automate 1 (CLI) 116
consultez aussi 117
Démarrage 117
Prérequis 117
Création d'un serveur Chef Automate 120
Fin de la configuration et chargement des livres de recettes 133
Ajout de nœuds à gérer 144
Connexion au tableau de bord Chef Automate 151
Créez un serveur Chef Automate dans CloudFormation 153
Prérequis 154
Créer un serveur Chef Automate dans AWS CloudFormation 155
Mise à jour d'un serveur pour utiliser un domaine personnalisé 162
Prérequis 163
Limites 53
Mise à jour d'un serveur pour utiliser un domaine personnalisé 54
consultez aussi 58
Régénérez le kit de démarrage 168
Régénérez le kit de AWS OpsWorks for Chef Automate démarrage avec AWS CLI 169
Utilisation des balises 170
Comment fonctionnent les tags dans AWS OpsWorks for Chef Automate 171
Ajouter et gérer des balises dans AWS OpsWorks for Chef Automate (console) 173
Ajouter et gérer des balises dans AWS OpsWorks for Chef Automate (CLI) 176

consultez aussi ... 181
Sauvegarde et restauration de serveurs ... 182
Sauvegarder un AWS OpsWorks for Chef Automate serveur ... 182
Restaurer un AWS OpsWorks for Chef Automate serveur ... 185
Maintenance du système ... 187
S'assurer que les nœuds font confiance à l'autorité AWS OpsWorks de certification ... 188
Configuration de la maintenance du système ... 189
Démarrage de la maintenance du système à la demande ... 191
Restauration des configurations personnalisées et des fichiers après la maintenance ... 192
Analyses de conformité ... 192
Conformité dans Chef Automate 2.0 ... 193
Conformité dans Chef Automate 1.x ... 201
Mises à jour de la conformité ... 207
Profils de conformité personnalisés et de la communauté ... 207
consultez aussi ... 208
Suppression de nœuds ... 208
Rubriques connexes ... 209
Suppression d'un serveur Chef Automate ... 210
Étape 1 : Dissociation de nœuds gérés ... 210
Étape 2 : Suppression du serveur ... 211
Réinitialiser les informations d'identification de Chef ... 211
En utilisant AWS CloudTrail ... 212
AWS OpsWorks for Chef Automate Informations dans CloudTrail ... 213
Comprendre les entrées du fichier AWS OpsWorks for Chef Automate journal ... 214
Résolution des problèmes ... 216
Conseils généraux de résolution des problèmes ... 217
Résolution d'erreurs spécifiques ... 217
Aide supplémentaire et support ... 225
Sécurité dans la gestion des AWS OpsWorks configurations (CM) ... 226
Protection des données ... 227
Intégration avec AWS Secrets Manager ... 229
Chiffrement des données ... 229
Chiffrement au repos ... 229
Chiffrement en transit ... 229
Gestion des clés ... 230
Gestion de l'identité et des accès ... 230

Public ciblé 230
Authentification avec des identités 231
Gestion des accès à l'aide de politiques 235
Comment fonctionne AWS OpsWorks CM avec IAM 238
Exemples de politiques basées sur l'identité 243
Résolution des problèmes 247
Politiques gérées par AWS 249
Prévention interservices confuse des adjoints dans AWS OpsWorks CM 258
Confidentialité du trafic inter-réseaux 262
Journalisation et surveillance 262
Validation de la conformité 262
Résilience 263
Sécurité de l'infrastructure 264
Configuration et analyse des vulnérabilités 265
Bonnes pratiques de sécurité 265
AWS OpsWorks Piles 267
Piles 270
Couches 271
Recettes et LifeCycle événements 271
instances 272
Applications 273
Personnalisation de votre pile 274
Gestion des ressources 275
Sécurité et autorisations 275
Surveillance et journalisation 276
CLI, SDK et modèles AWS CloudFormation 276
Fin de vie FAQs 277
Comment les clients existants seront-ils affectés par cette fin de vie ? 277
Est-ce AWS OpsWorks Stacks accepter de nouveaux clients ? 278
Où dois-je migrer mes piles existantes ? 278
Comment puis-je conserver mes EC2 instances Amazon existantes après leur fin de vie ? . 278
La fin de vie aura-t-elle un impact sur Régions AWS tout le monde en même temps ? 279
Quel est le niveau de support technique disponible AWS OpsWorks Stacks ? 279
Y aura-t-il de nouvelles fonctionnalités pour AWS OpsWorks Stacks ? 279
Migration de vos applications vers Systems Manager Application Manager 279
Comment fonctionne le script 280

Prérequis ... 280
Limites ... 281
Premiers pas ... 282
FAQ ... 298
Résolution des problèmes ... 310
Utilisation de l' AWS OpsWorks Stacks outil Detach in Place ... 311
Comment fonctionne le processus ... 312
Limites ... 315
Premiers pas ... 316
Démarrage ... 326
Prise en charge de la région ... 327
Mise en route : exemple ... 328
Mise en route : Linux ... 350
Mise en route : Windows ... 382
Mise en route : livres de recettes ... 419
Bonnes pratiques ... 455
Stockage de périphérique racine ... 456
Optimisation du nombre de serveurs ... 458
Gestion des autorisations ... 461
Gestion et déploiement des applications et livres de recettes ... 465
Empaquetage local des dépendances des livres de recettes ... 474
Piles ... 479
Migrer des piles depuis -Classic EC2 ... 481
Créer une pile ... 484
Running a Stack in a VPC ... 493
Mise à jour d'une pile ... 505
Cloner une pile ... 506
Exécution des commandes de pile ... 508
Utilisation du JSON personnalisé ... 511
Suppression d'une pile ... 514
Couches ... 519
OpsWorks Principes de base des couches ... 520
Couche Elastic Load Balancing ... 537
Couche de service Amazon RDS ... 542
Couches de cluster ECS ... 548
Couches personnalisées ... 555

Installations de package par couche 556
instances 558
Utilisation des AWS OpsWorks instances Stacks 559
Utilisation des ressources de calcul créées en dehors d' AWS OpsWorks Stacks 623
Modification de la configuration de l'instance 673
Supprimer des AWS OpsWorks instances de Stacks 675
Connexion avec SSH 677
Connexion avec RDP 680
Applications 685
Ajout d'applications 686
Déploiement d'applications 694
Modification des applications 698
Connexion à une base de données 700
Utilisation des variables d'environnement 702
Transmission de données aux applications 704
Utilisation de clés SSH d'un référentiel Git 708
Utilisation des domaines personnalisés 709
Utilisation de SSL 711
Livres de recettes et recettes 719
Référentiels de livres de recettes 720
Versions Chef 724
Versions de Ruby 744
Installation de livres de recettes personnalisés 746
Mise à jour des livres de recettes personnalisés 750
Exécution des recettes 752
Gestion des ressources 761
Enregistrement des ressources auprès d'une pile 763
Attachement et déplacement de ressources 769
Détachement des ressources 774
Annulation de l'enregistrement des ressources 776
Balises 779
Configuration des balises au niveau de la pile 780
Configuration des balises au niveau d'une couche 782
Gérer les tags à l'aide du AWS CLI 784
Limitations des balises 785
Surveillance 786

Utilisation d'Amazon CloudWatch 786
En utilisant AWS CloudTrail 799
Utilisation d'Amazon CloudWatch Logs 802
Utilisation d'Amazon CloudWatch Events 808
Sécurité et autorisations 809
Gestion des autorisations utilisateur 810
Permettre à AWS OpsWorks Stacks d'agir en votre nom 836
Prévention de l'adjoint confus 841
Spécification des autorisations pour les applications exécutées sur EC2 des instances 845
Gestion de l'accès SSH 849
Gestion des mises à jour de sécurité 857
Utilisation des groupes de sécurité 859
Chef 12 Linux 862
Présentation 863
Migration vers Chef 12 864
Systèmes d'exploitation pris en charge 866
Types d'instance pris en charge 866
En savoir plus 866
Passage aux conteneurs de données 867
Versions Chef antérieures 869
Chef 11.10 et versions antérieures pour Linux 869
Utilisation de AWS OpsWorks Stacks avec d'autres services AWS 1320
Utilisation d'un magasin de données principal 1321
ElastiCache Redis 1330
Utilisation d'un compartiment Amazon S3 1345
Utilisation AWS CodePipeline avec AWS OpsWorks Stacks 1359
Utilisation de la AWS OpsWorks CLI Stacks 1423
Créer une instance 1426
Déployer une application 1429
Répertorier les applications 1430
Commandes de liste 1431
Répertorier les déploiements 1433
Répertorier les adresses IP Elastic 1434
Répertorier les instances 1435
Répertorier les piles 1436
Répertorier les couches 1438

Exécuter une recette 1442
Install Dependencies 1443
Mettre à jour la configuration de pile 1444
Guide de débogage et dépannage 1445
Débogage des recettes 1446
Débogage et dépannage des problèmes courants 1465
AWS OpsWorks CLI de l'agent Stacks 1475
agent_report 1478
get_json 1478
instance_report 1483
list_commands 1484
run_command 1485
show_log 1486
stack_state 1487
AWS OpsWorks Référence du sac de données Stacks 1490
Conteneur de données d'application (aws_opsworks_app) 1494
Conteneurs de données de commandes (aws_opsworks_command) 1498
Sac de données du cluster Amazon ECS (aws_opsworks_ecs_cluster) 1500
Sac de données Elastic Load Balancing (aws_opsworks_elastic_load_balancer) 1501
Conteneur de données de l'instance (aws_opsworks_instance) 1502
Conteneur de données de la couche (aws_opsworks_layer) 1508
Sac de données Amazon RDS (aws_opsworks_rds_db_instance) 1510
Conteneur de données de la pile (aws_opsworks_stack) 1511
Conteneur de données utilisateur (aws_opsworks_user) 1513
OpsWorks Changements d'agent 1515
Versions de l'agent Chef 12 1515
Versions de l'agent Chef 11.10 1519
Ressources 1526
Guides de référence, outils et ressources de prise en charge 1526
AWS Kits de développement logiciel 1527
Logiciels Open Source 1528
AWS OpsWorks historique du document 1529
Mises à jour antérieures 1537
..... mdxli

Qu'est-ce que c'est AWS OpsWorks ?

> **Important**
>
> Les AWS OpsWorks services ont atteint la fin de leur durée de vie et ont été désactivés pour les nouveaux clients comme pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks est un service de gestion de configuration qui vous aide à configurer et à exploiter des applications dans une entreprise cloud à l'aide de Puppet ou Chef. AWS OpsWorks Permet de cumuler et AWS OpsWorks for Chef Automate d'utiliser les livres de recettes et les solutions Chef pour la gestion de la configuration, tandis que OpsWorks pour Puppet Enterprise, vous pouvez configurer un serveur principal Puppet Enterprise dans. AWS Puppet vous donne accès à un ensemble d'outils pour appliquer l'état souhaité à votre infrastructure et automatiser les tâches à la demande.

AWS OpsWorks Des services

AWS OpsWorks pour Puppet Enterprise

OpsWorks for Puppet Enterprise vous permet de créer des serveurs AWS maîtres Puppet gérés. Un serveur maître Puppet gère les nœuds de votre infrastructure, stocke des données concernant ces nœuds et fait office de référentiel centralisé pour vos modules Puppet. Les modules sont des unités de code Puppet partageables et réutilisables qui contiennent des instructions sur la façon dont votre infrastructure doit être configurée. Vous pouvez télécharger des modules proposés par la communauté à partir de Puppet Forge, ou utiliser le kit de développement Puppet pour créer vos propres modules personnalisés, puis gérer leur déploiement avec Puppet Code Manager.

OpsWorks for Puppet Enterprise fournit un Puppet master entièrement géré, une suite d'outils d'automatisation qui vous permettent d'inspecter, de livrer, d'exploiter et de pérenniser vos applications, ainsi qu'un accès à une interface utilisateur qui vous permet de consulter les informations relatives à vos nœuds et aux activités de Puppet. OpsWorks for Puppet Enterprise vous permet d'utiliser Puppet pour automatiser la façon dont les nœuds sont configurés, déployés

et gérés, qu'il s'agisse d' EC2 instances Amazon ou d'appareils sur site. An OpsWorks for Puppet Enterprise master fournit une automatisation complète en gérant des tâches telles que la configuration des logiciels et des systèmes d'exploitation, les installations de packages, les configurations de bases de données, la gestion des modifications, l'application des politiques, la surveillance et l'assurance qualité.

Comme OpsWorks Puppet Enterprise gère le logiciel Puppet Enterprise, votre serveur peut être sauvegardé automatiquement au moment de votre choix, exécute toujours la version compatible AWS la plus récente de Puppet et dispose toujours des mises à jour de sécurité les plus récentes. Vous pouvez utiliser les groupes Amazon EC2 Auto Scaling pour associer automatiquement les nouveaux EC2 nœuds Amazon à votre serveur.

AWS OpsWorks pour Chef Automate

AWS OpsWorks for Chef Automate vous permet de créer des serveurs Chef AWS gérés qui incluent les fonctionnalités premium de Chef Automate et d'utiliser le Chef DK et d'autres outils Chef pour les gérer. Un serveur Chef gère les nœuds de votre environnement, stocke des informations sur ces nœuds et fait office de référentiel centralisé pour vos livres de recettes Chef. Les livres de recettes contiennent les recettes exécutées par l'agent du client Chef Infra (`chef-client`) sur chaque nœud que vous gérez à l'aide de Chef. Vous pouvez utiliser des outils Chef tels que `knife`Test Kitchen pour gérer les nœuds et les livres de recettes sur un serveur Chef du AWS OpsWorks for Chef Automate service.

Chef Automate est un progiciel de serveur inclus qui fournit un flux de travail automatisé pour le déploiement continu et les contrôles de conformité. AWS OpsWorks for Chef Automate installe et gère Chef Automate, Chef Infra et Chef à l'aide InSpec d'une seule instance Amazon Elastic Compute Cloud. Avec AWS OpsWorks for Chef Automate, vous pouvez utiliser des livres de cuisine Chef créés par la communauté ou personnalisés sans apporter AWS OpsWorks de modifications spécifiques.

Comme il AWS OpsWorks for Chef Automate gère les composants de Chef Automate sur une seule instance, votre serveur peut être sauvegardé automatiquement au moment de votre choix, exécute toujours la version mineure la plus récente de Chef et dispose toujours des mises à jour de sécurité les plus récentes. Vous pouvez utiliser les groupes Amazon EC2 Auto Scaling pour associer automatiquement les nouveaux EC2 nœuds Amazon à votre serveur.

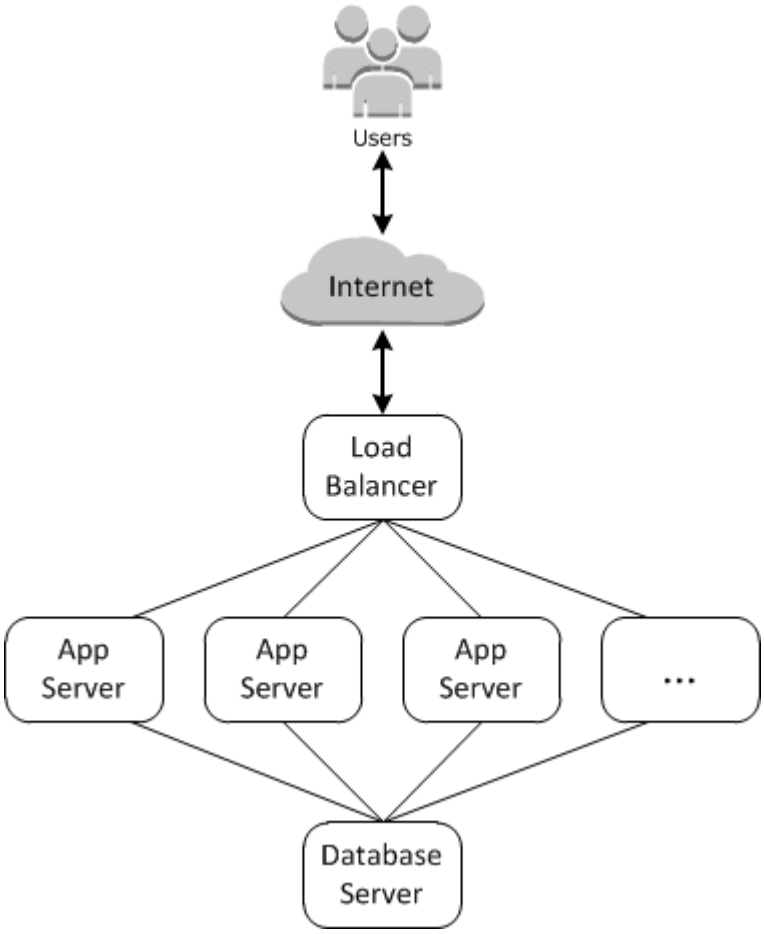
AWS OpsWorks Piles

L'informatique basée sur le cloud implique généralement des groupes de ressources AWS, tels que EC2 des instances et des instances Amazon Relational Database Service (RDS). Par

exemple, une application web nécessite généralement des serveurs d'applications, des serveurs de base des données, des équilibreurs de charge et d'autres ressources. Ce groupe d'instances est généralement appelé une pile.

AWS OpsWorks Stacks, le service original, fournit un moyen simple et flexible de créer et de gérer des piles et des applications. AWS OpsWorks Stacks vous permet de déployer et de surveiller les applications de vos stacks. Vous pouvez créer des piles qui vous aideront à gérer les ressources cloud dans des groupes spécialisés appelés couches. Une couche représente un ensemble d' EC2 instances qui répondent à un objectif particulier, comme le service d'applications ou l'hébergement d'un serveur de base de données. Les couches dépendent des recettes Chef pour gérer des tâches telles que l'installation de packages sur les instances, le déploiement d'applications et l'exécution de scripts.

Contrairement à AWS OpsWorks for Chef Automate AWS OpsWorks Stacks ne nécessite ni ne crée de serveurs Chef ; AWS OpsWorks Stacks effectue une partie du travail d'un serveur Chef à votre place. AWS OpsWorks Stacks surveille l'état de santé des instances et fournit de nouvelles instances pour vous, si nécessaire, en utilisant Auto Healing et Auto Scaling. Une pile simple de serveurs d'applications peut ressembler au schéma suivant.



AWS OpsWorks pour Puppet Enterprise

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

OpsWorks for Puppet Enterprise vous permet de lancer un master Puppet Enterprise en quelques minutes et de AWS OpsWorks gérer ses opérations, ses sauvegardes, ses restaurations et ses mises à niveau logicielles. OpsWorks for Puppet Enterprise vous permet de vous concentrer sur les tâches de gestion de configuration de base, au lieu de gérer un Puppet Master. En utilisant OpsWorks for Puppet Enterprise, vous pouvez utiliser les mêmes configurations pour gérer à la fois votre infrastructure sur site et dans le cloud, ce qui vous permet de faire évoluer efficacement vos opérations dans un environnement hybride. La gestion de votre serveur maître Puppet est simplifiée par la console Puppet Enterprise, l' AWS Management Console et l' AWS CLI.

Un maître Puppet gère la configuration des nœuds dans votre environnement en diffusant des catalogues de configuration pour certains nœuds au logiciel `puppet-agent` et sert de référentiel central pour vos modules Puppet. Un Puppet Master intégré OpsWorks à Puppet Enterprise se déploie `puppet-agent` sur vos nœuds gérés et fournit les fonctionnalités premium de Puppet Enterprise.

Un master OpsWorks for Puppet Enterprise s'exécute sur une instance Amazon Elastic Compute Cloud. OpsWorks pour les serveurs Puppet Enterprise sont configurés pour exécuter la dernière version d'Amazon Linux (Amazon Linux 2) et la version la plus récente de Puppet Enterprise Master, version 2019.8.5. Pour plus d'informations sur les modifications apportées à Puppet Enterprise 2019.8.5, consultez les notes de mise à jour de Puppet Enterprise.

Lorsque de nouvelles versions du logiciel Puppet deviennent disponibles, la maintenance système est conçue pour mettre à jour automatiquement la version de Puppet Enterprise sur le serveur dès que celle-ci réussit les tests AWS. AWS exécute des tests approfondis pour vérifier que les mises à niveau Puppet sont prêtes à être déployées en production et ne perturbent pas les environnements client existants.

Vous pouvez connecter n'importe quel ordinateur ou EC2 instance sur site exécutant un système d'exploitation compatible et disposant d'un accès réseau à un master OpsWorks pour Puppet Enterprise. Le maître Puppet installe le logiciel agent `puppet` sur les nœuds que vous voulez gérer.

Rubriques

- Support régional OpsWorks pour Puppet Enterprise
- AWS OpsWorks for Puppet Enterprise Fin de vie FAQs
- Commencer à utiliser OpsWorks pour Puppet Enterprise
- Créez un AWS OpsWorks for Puppet Enterprise Master en utilisant AWS CloudFormation
- Mettre à jour un serveur OpsWorks pour Puppet Enterprise afin d'utiliser un domaine personnalisé
- Utilisation des balises sur les AWS OpsWorks for Puppet Enterprise ressources
- Sauvegarder et restaurer un serveur OpsWorks pour Puppet Enterprise
- Maintenance du système dans OpsWorks Puppet Enterprise
- Ajout automatique de nœuds dans OpsWorks Puppet Enterprise
- Dissocier un nœud d'un serveur OpsWorks for Puppet Enterprise
- Supprimer un OpsWorks pour Puppet Enterprise Server
- Comment migrer un serveur OpsWorks pour Puppet Enterprise vers Amazon Elastic Compute Cloud (Amazon EC2)
- Journalisation OpsWorks des appels d'API Puppet Enterprise avec AWS CloudTrail
- Résolution des problèmes OpsWorks pour Puppet Enterprise

Support régional OpsWorks pour Puppet Enterprise

Les points de terminaison régionaux suivants sont compatibles avec OpsWorks les masters de Puppet Enterprise. OpsWorks for Puppet Enterprise crée des ressources associées à vos Puppet masters, telles que des profils d'instance, des utilisateurs et des rôles de service, sur le même point de terminaison régional que votre Puppet master. Votre maître Puppet doit se trouver dans un VPC. Vous pouvez utiliser un VPC que vous créez ou que vous possédez déjà, ou utiliser le VPC par défaut.

- Région US East (Ohio)
- Région US East (N. Virginia)
- Région US West (N. California)

- Région US West (Oregon)
- Région Asie-Pacifique (Tokyo)
- Région Asia Pacific (Singapore)
- Région Asia Pacific (Sydney)
- Région Europe (Frankfurt)
- Europe (Ireland) Region

AWS OpsWorks for Puppet Enterprise Fin de vie FAQs

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Rubriques

- Comment les clients existants seront-ils affectés par cette fin de vie ?
- Qu'arrive-t-il à mes serveurs si je n'agis pas ?
- Est-ce AWS OpsWorks for Puppet Enterprise accepter de nouveaux clients ?
- La fin de vie aura-t-elle un impact sur Régions AWS tout le monde en même temps ?
- Quel est le niveau de support technique disponible AWS OpsWorks for Puppet Enterprise ?
- Je suis actuellement client de OpsWorks for Puppet Enterprise et je dois lancer un serveur sur un compte qui n'utilisait pas le service auparavant. Suis-je capable de le faire ?
- Y aura-t-il de nouvelles fonctionnalités pour AWS OpsWorks for Puppet Enterprise ?

Comment les clients existants seront-ils affectés par cette fin de vie ?

Les clients existants ne seront pas concernés jusqu'au 31 mars 2024, date de fin de vie de Puppet Enterprise. OpsWorks Après la date de fin de vie, les clients ne seront plus en mesure de gérer leurs serveurs à l'aide de la OpsWorks console ou de l'API.

Qu'arrive-t-il à mes serveurs si je n'agis pas ?

À compter du 31 mars 2024, vous ne pourrez plus gérer vos serveurs à l'aide de la OpsWorks console ou de l'API. À ce moment-là, nous cesserons d'exécuter les fonctions de gestion continue de vos serveurs, telles que les sauvegardes ou la maintenance. Pour limiter l'impact sur les clients, nous laisserons fonctionner les EC2 instances qui sauvegardent les serveurs Puppet Enterprise, mais leurs licences ne seront plus valides car l'utilisation n'est plus couverte (ou facturée) dans le cadre du contrat de service OpsWorks pour Puppet Enterprise. Si vous souhaitez continuer à gérer votre infrastructure avec Puppet Enterprise, consultez Comment migrer un OpsWorks serveur Puppet Enterprise vers Amazon Elastic Compute Cloud (Amazon EC2).

Est-ce AWS OpsWorks for Puppet Enterprise accepter de nouveaux clients ?

Non AWS OpsWorks for Puppet Enterprise n'accepte plus de nouveaux clients.

La fin de vie aura-t-elle un impact sur Régions AWS tout le monde en même temps ?

Oui. L'API et la console atteindront leur fin de vie et seront inutilisables à compter du 31 mars 2024 dans toutes les régions. Pour une liste des Régions AWS endroits où ces services AWS OpsWorks for Puppet Enterprise sont disponibles, consultez la liste des services AWS régionaux.

Quel est le niveau de support technique disponible AWS OpsWorks for Puppet Enterprise ?

AWS continuera à fournir le même niveau de support AWS OpsWorks for Puppet Enterprise que celui dont bénéficient les clients aujourd'hui jusqu'à la date de fin de vie. Si vous avez des questions ou des préoccupations, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Je suis actuellement client de OpsWorks for Puppet Enterprise et je dois lancer un serveur sur un compte qui n'utilisait pas le service auparavant. Suis-je capable de le faire ?

En général, non, sauf dans des circonstances exceptionnelles. Si vous vous trouvez dans une situation particulière, contactez l' AWS Support équipe sur AWS Re:post ou via le AWS Support

Premium en fournissant les détails et les justifications nécessaires, et nous examinerons votre demande.

Y aura-t-il de nouvelles fonctionnalités pour AWS OpsWorks for Puppet Enterprise ?

Non Le service étant sur le point d'atteindre sa fin de vie, nous ne publierons aucune nouvelle fonctionnalité. Cependant, nous continuerons à améliorer la sécurité et à gérer les serveurs comme prévu jusqu'à la date de fin de vie.

Commencer à utiliser OpsWorks pour Puppet Enterprise

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

OpsWorks pour Puppet Enterprise vous permet d'exécuter un serveur Puppet Enterprise dans. AWS Il ne faut que 15 minutes environ pour mettre en service un serveur maître Puppet Enterprise.

À compter du 3 mai 2021, OpsWorks for Puppet Enterprise stocke certains attributs du serveur Puppet Enterprise dans. AWS Secrets Manager Pour de plus amples informations, veuillez consulter Intégration avec AWS Secrets Manager.

La procédure pas à pas suivante vous aide à créer votre premier Puppet Master dans Puppet OpsWorks Enterprise.

Prérequis

Avant de commencer, vous devez remplir les conditions préalables suivantes.

Rubriques

- Installer le kit de développement Puppet

- Installer les outils du client Puppet Enterprise
- Configurer un référentiel de contrôle Git
- Configuration d'un VPC
- Configurer une paire de EC2 clés (facultatif)
- Prérequis pour l'utilisation d'un domaine personnalisé (facultatif)

Installer le kit de développement Puppet

1. Depuis le site web de Puppet, téléchargez le kit de développement Puppet correspondant au système d'exploitation de votre ordinateur local.
2. Installez le kit de développement Puppet.
3. Ajoutez le kit de développement Puppet à la variable PATH de votre ordinateur local.

   - Sur un système d'exploitation Linux ou MacOS, vous pouvez ajouter le kit de développement Puppet à votre variable PATH en exécutant la commande ci-dessous dans un shell Bash.

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

   - Sur un système d'exploitation Windows, vous pouvez ajouter le kit de développement Puppet à votre PATH variable en utilisant la commande .NET Framework suivante dans une PowerShell session ou dans la boîte de dialogue des variables d'environnement accessible depuis les propriétés du système. Vous devrez peut-être exécuter votre PowerShell session en tant qu'administrateur pour exécuter la commande suivante.

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

Installer les outils du client Puppet Enterprise

Les outils du client Puppet Enterprise (PE) constituent un ensemble d'outils de ligne de commande qui vous permettent d'accéder aux services Puppet Enterprise depuis votre poste de travail. Ils peuvent être installés sur de nombreux systèmes d'exploitation différents, ainsi que sur les nœuds que vous gérez à l'aide de Puppet. Pour plus d'informations sur les systèmes d'exploitation pris en charge pour ces outils, ainsi que la façon de les installer, consultez la section Installing PE client tools (Installation d'outils client PE) dans la documentation de Puppet Enterprise.

Configurer un référentiel de contrôle Git

Avant de pouvoir lancer un serveur maître Puppet, un référentiel de contrôle doit être configuré dans Git pour pouvoir stocker et modifier/gérer vos modules et classes Puppet. Pour permettre le lancement du serveur maître Puppet Enterprise, vous avez besoin d'une URL vers un référentiel Git et d'informations de compte SSH ou HTTPS pour accéder au référentiel. Pour savoir comment configurer un référentiel de contrôle utilisable par votre serveur maître Puppet Enterprise, consultez la section Setting up a control repository (Configuration d'un référentiel de contrôle). Vous pouvez également trouver les instructions de configuration du référentiel de contrôle dans le fichier readme du référentiel d'`control-repo`échantillons de Puppet sur. GitHub La structure du référentiel de contrôle ressemble à ce qui suit.

```
### 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
```

Configuration d'un référentiel à l'aide de CodeCommit

Vous pouvez créer un nouveau dépôt en utilisant CodeCommit. Pour plus d'informations sur la façon CodeCommit de créer votre référentiel de contrôle, consultez the section called “Facultatif : utilisation

CodeCommit" ce guide. Pour plus d'informations sur la façon de démarrer avec Git on CodeCommit, consultez Getting started with AWS CodeCommit. Pour autoriser votre OpsWorks serveur Puppet Enterprise à accéder à votre référentiel, associez la `AWSCodeCommitReadOnly` politique à votre rôle de profil d'instance IAM.

Configuration d'un VPC

Votre master OpsWorks for Puppet Enterprise doit fonctionner dans un Amazon Virtual Private Cloud. Vous pouvez l'ajouter à un VPC existant, utiliser le VPC par défaut, ou créer un VPC pour contenir le serveur. Pour plus d'informations sur Amazon VPC et sur la façon de créer un nouveau VPC, consultez le guide de démarrage Amazon VPC.

Si vous créez votre propre VPC ou si vous utilisez un VPC existant, celui-ci doit avoir les propriétés ou paramètres suivants.

- Le VPC doit avoir au moins un sous-réseau.

  Si votre master OpsWorks for Puppet Enterprise doit être accessible au public, rendez le sous-réseau public et activez l'attribution automatique d'une adresse IP publique.
- L'option Résolution DNS doit être activée.
- Sur le sous-réseau, activez Attribuer automatiquement l'adresse IP publique.

Configurer une paire de EC2 clés (facultatif)

Une connexion SSH n'est ni nécessaire ni recommandée pour la gestion classique du serveur Puppet ; vous pouvez utiliser les AWS CLI commandes AWS Management Console et pour effectuer de nombreuses tâches de gestion sur votre serveur Puppet.

Une paire de EC2 clés est requise pour vous connecter à votre serveur en utilisant SSH au cas où vous perdriez ou souhaiteriez modifier le mot de passe de connexion à la console Web de Puppet Enterprise. Vous pouvez utiliser une paire de clés existante, ou en créer une. Pour plus d'informations sur la création d'une nouvelle paire de EC2 clés, consultez Amazon EC2 Key Pairs.

Si vous n'avez pas besoin d'une paire de EC2 clés, vous êtes prêt à créer un master Puppet Enterprise.

Prérequis pour l'utilisation d'un domaine personnalisé (facultatif)

Vous pouvez configurer votre maître Chef Automate sur votre propre domaine, en spécifiant un point de terminaison public dans un domaine personnalisé à utiliser comme point de terminaison de votre

serveur. Lorsque vous utilisez un domaine personnalisé, tous les éléments et opérations ci-après sont requis, comme décrit en détail dans cette section.

Rubriques

- Configurer un domaine personnalisé
- Obtenir un certificat
- Obtenir une clé privée

Configurer un domaine personnalisé

Pour exécuter votre maître Chef Automate sur votre propre domaine personnalisé, vous avez besoin d'un point de terminaison public d'un serveur, tel que `https://aws.my-company.com`. Si vous spécifiez un domaine personnalisé, vous devez également fournir un certificat et une clé privée, comme décrit dans les sections précédentes.

Pour accéder au serveur après sa création, ajoutez un enregistrement DNS CNAME dans votre service DNS préféré. Cet enregistrement doit faire pointer le domaine personnalisé vers le point de terminaison (la valeur de l'attribut `Endpoint` du serveur) généré par le processus de création du maître Puppet. Vous ne pouvez pas accéder au serveur à l'aide de la valeur `Endpoint` générée si le serveur utilise un domaine personnalisé.

Obtenir un certificat

Pour configurer votre maître Puppet sur votre propre domaine personnalisé, vous avez besoin d'un certificat HTTPS au format PEM. Il peut s'agir d'un seul certificat auto-signé ou d'une chaîne de certificats. Lorsque vous effectuez le flux de travail Créer un maître Puppet Enterprise, si vous spécifiez ce certificat, vous devez également fournir un domaine et une clé privée personnalisés.

Voici les conditions requises pour la valeur du certificat :

- Vous pouvez fournir un certificat personnalisé auto-signé ou la chaîne de certificats complète.
- Le certificat doit être un certificat X509 valide ou une chaîne de certificats au format PEM.
- Le certificat doit être valide au moment du chargement. Vous ne pouvez pas importer un certificat avant le début de sa période de validité (date `NotBefore` du certificat) ou après son expiration (date `NotAfter` du certificat).
- Le nom commun ou les noms alternatifs du sujet (SANs) du certificat, s'ils sont présents, doivent correspondre à la valeur de domaine personnalisée.

- Le certificat doit correspondre à la valeur du champ Custom private key (Clé privée personnalisée)
  .

Obtenir une clé privée

Pour configurer votre maître Puppet sur votre propre domaine personnalisé, vous avez besoin d'une clé privée au format PEM pour vous connecter au serveur à l'aide du protocole HTTPS. La clé privée ne doit pas être chiffrée ; elle ne peut pas être protégée par un mot de passe ou une phrase passe. Si vous spécifiez une clé privée personnalisée, vous devez également fournir un domaine et un certificat personnalisés.

Créer un serveur maître Puppet Enterprise

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

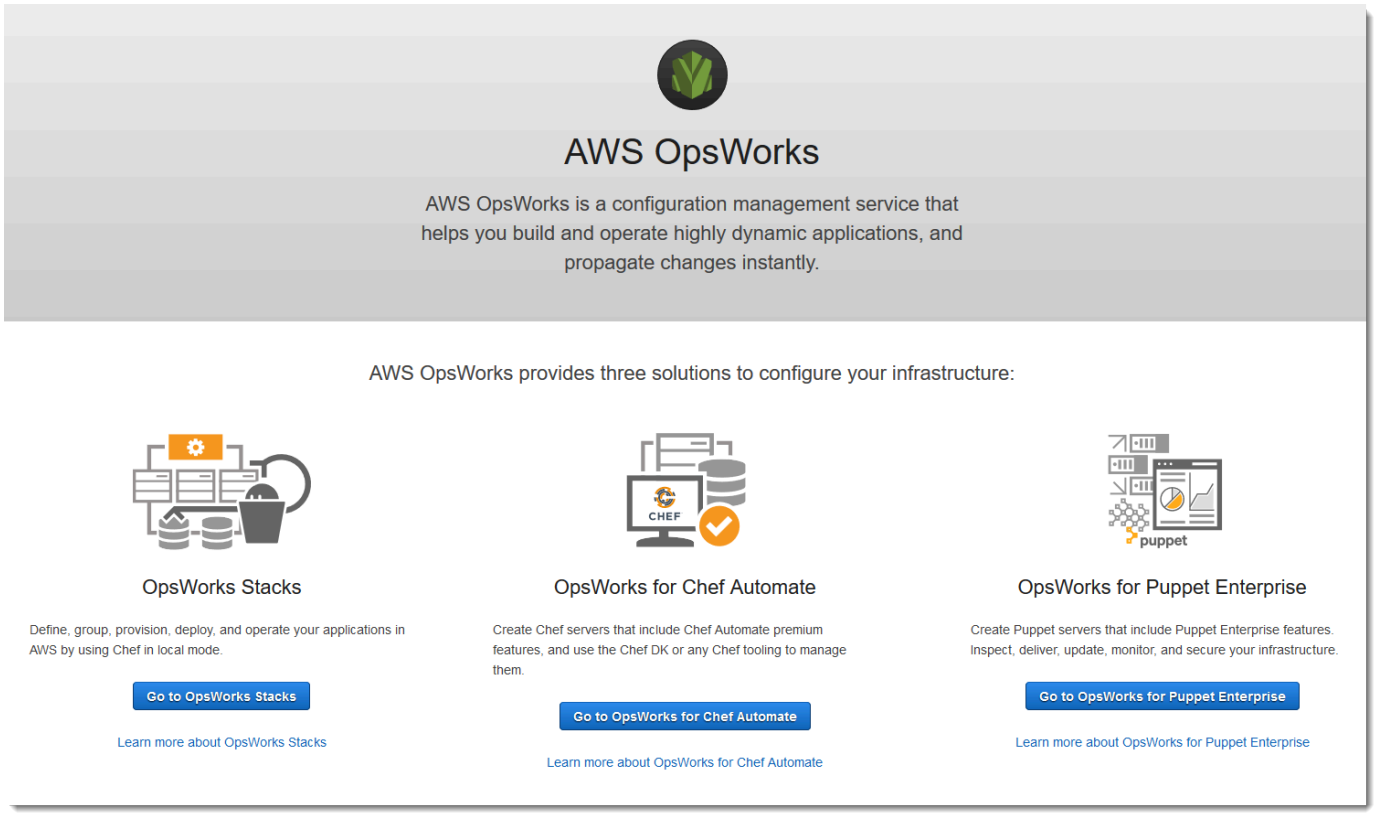
Vous pouvez créer un Puppet Master à l'aide de la console OpsWorks for Puppet Enterprise ou du. AWS CLI

Rubriques

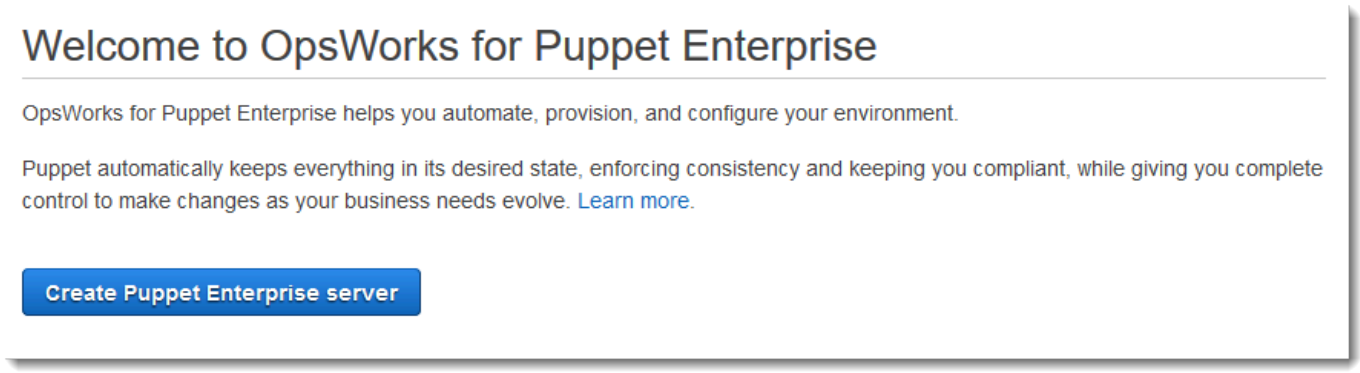
- Créez un Puppet Enterprise Master à l'aide du AWS Management Console
- Créez un Puppet Enterprise Master à l'aide du AWS CLI

Créez un Puppet Enterprise Master à l'aide du AWS Management Console

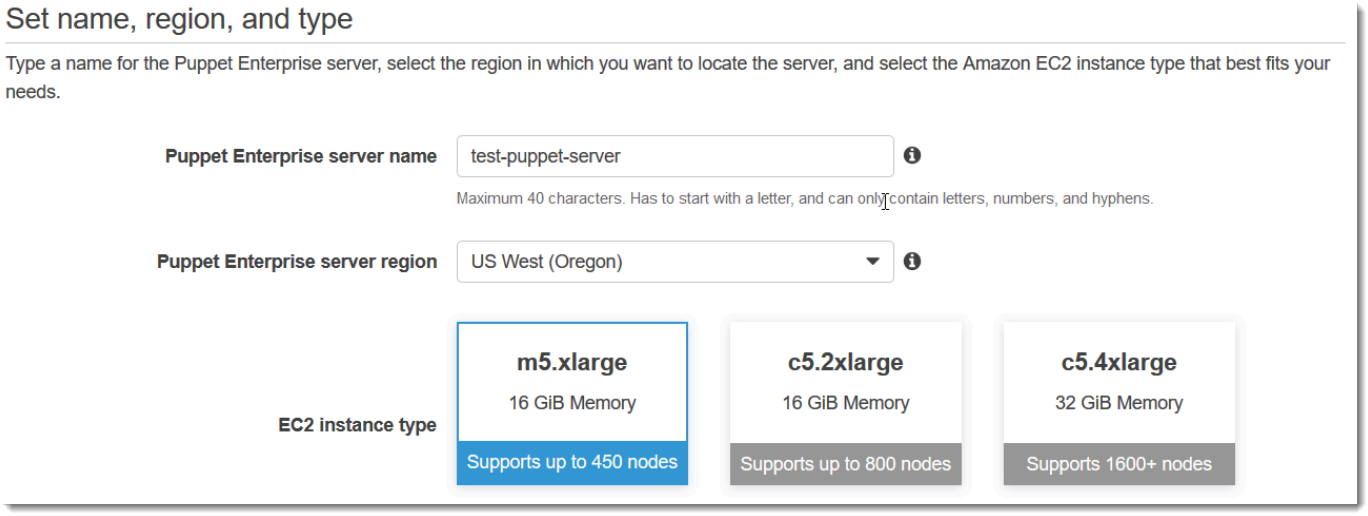
1. Connectez-vous à la AWS OpsWorks console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/opsworks/.
2. Sur la page d' AWS OpsWorks accueil, choisissez Go to OpsWorks pour Puppet Enterprise.



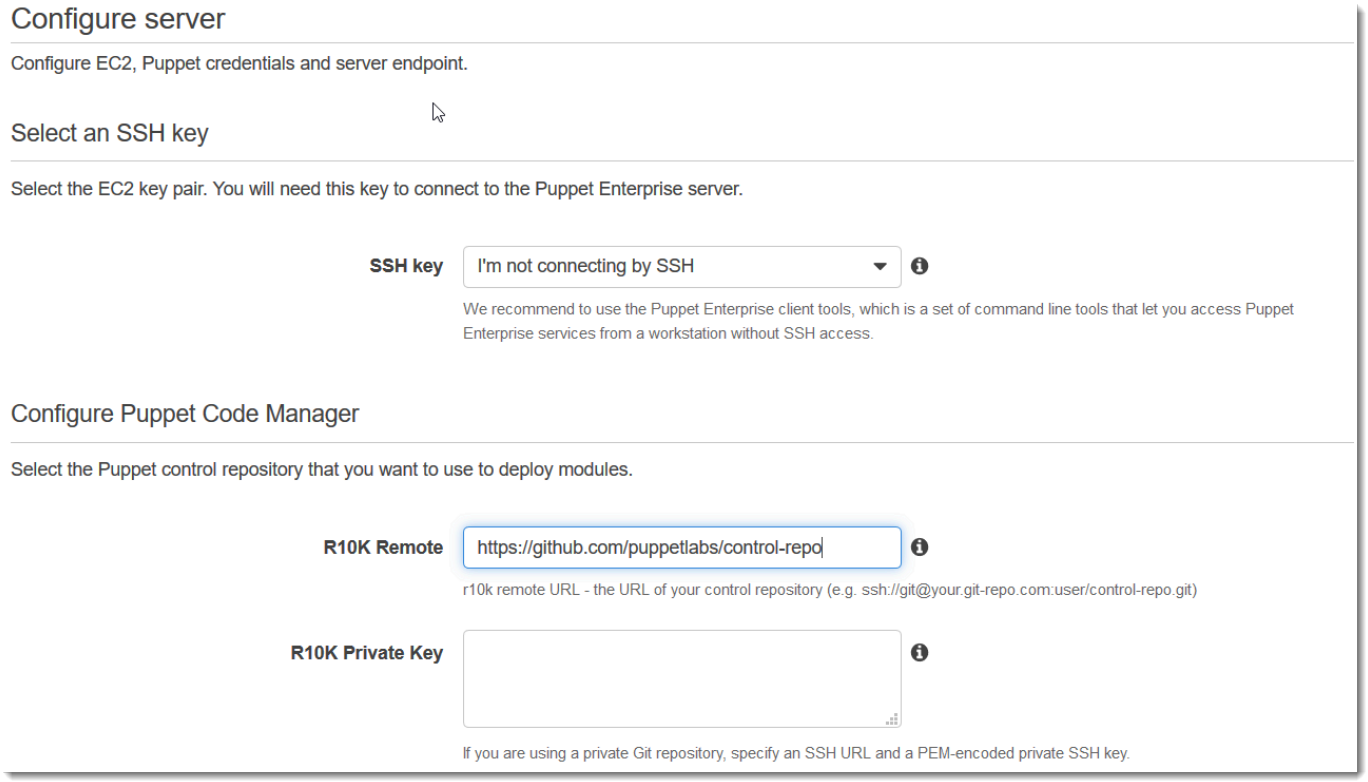
3. Sur la page OpsWorks d'accueil de Puppet Enterprise, choisissez Create Puppet Enterprise server.



4. Sur la page Set name, region, and type (Définir le nom, la région et le type), spécifiez un nom pour votre serveur. Les noms de serveur maître Puppet peuvent comporter 40 caractères au maximum, doivent commencer par une lettre et contenir uniquement des caractères alphanumériques et des tirets. Sélectionnez une région prise en charge, puis choisissez un type d'instance prenant en charge le nombre de nœuds que vous voulez gérer. Vous pouvez modifier le type d'instance une fois que votre serveur a été créé, si nécessaire. Pour cette procédure pas à pas, nous créons un type d'instance m5.xlarge dans la région USA Ouest (Oregon). Choisissez Suivant.



5. Sur la page Configure server (Configurer le serveur), conservez la sélection par défaut dans la liste déroulante SSH key (Clé SSH), sauf si vous souhaitez spécifier un nom de paire de clés. Dans le champ r10k remote de la zone Configure Puppet Code Manager (Configurer Puppet Code Manager), spécifiez une URL SSH ou HTTPS valide pour votre git remote. Dans le champ clé privée r10k, collez la clé privée SSH qui AWS OpsWorks peut être utilisée pour accéder au référentiel distant r10k. Celle-ci est fournie par Git lorsque vous créez un référentiel privé, mais n'est pas requise si vous utilisez l'authentification HTTPS pour accéder à votre référentiel de contrôle. Choisissez Suivant.

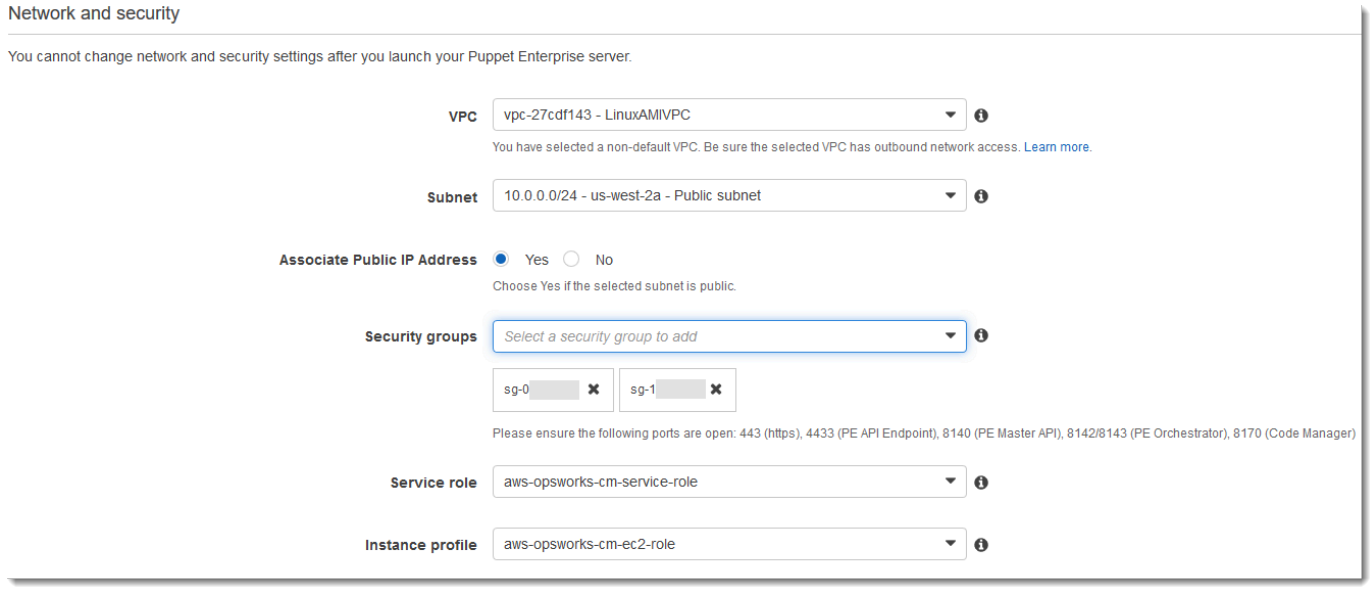


6. Pour Specify server endpoint (Spécifier le point de terminaison du serveur), laissez la valeur par défaut, Use an automatically-generated endpoint (Utiliser un point de terminaison généré automatiquement), puis choisissez Next (Suivant), sauf si vous souhaitez que votre serveur se trouve sur votre propre domaine personnalisé. Pour configurer un domaine personnalisé, passez à l'étape suivante.
7. Pour utiliser un domaine personnalisé, dans Specify server endpoint (Spécifier le point de terminaison du serveur), choisissez Utiliser un domaine personnalisé (Use a custom domain) dans la liste déroulante.

   a. Pour Fully qualified domain name (FQDN) (Nom de domaine complet (FQDN)), spécifiez un nom de domaine complet. Vous devez posséder le nom de domaine que vous souhaitez utiliser.

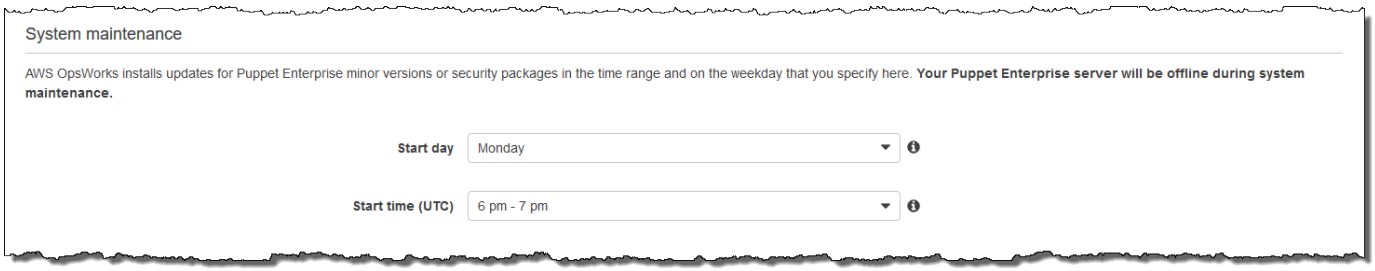
   b. Pour SSL certificate (Certificat SSL), collez l'intégralité du certificat au format PEM, en commençant par `-----BEGIN CERTIFICATE-----` et se terminant par `-----END CERTIFICATE-----`. L'objet du certificat SSL doit correspondre au nom de domaine complet que vous avez entré à l'étape précédente. Supprimez les lignes supplémentaires avant et après le certificat.

   c. Pour SSL private key (Clé privée SSL), collez l'intégralité de la clé privée RSA, en commençant par `-----BEGIN RSA PRIVATE KEY-----` et terminant par `-----END RSA PRIVATE KEY-----`. La clé privée SSL doit correspondre à la clé publique du certificat SSL que vous avez entré à l'étape précédente. Supprimez les lignes supplémentaires avant et après la clé privée. Choisissez Suivant.
8. Sur la page Configurer les paramètres avancés, dans la zone Réseau et sécurité, choisissez un VPC, un sous-réseau et un ou plusieurs groupes de sécurité. AWS OpsWorks peut générer un groupe de sécurité, un rôle de service et un profil d'instance pour vous, si vous n'en avez pas déjà un à utiliser. Votre serveur peut être membre de plusieurs groupes de sécurité. Une fois que vous avez quitté cette page, vous ne pouvez plus modifier les paramètres de réseau et de sécurité pour le serveur maître Puppet.

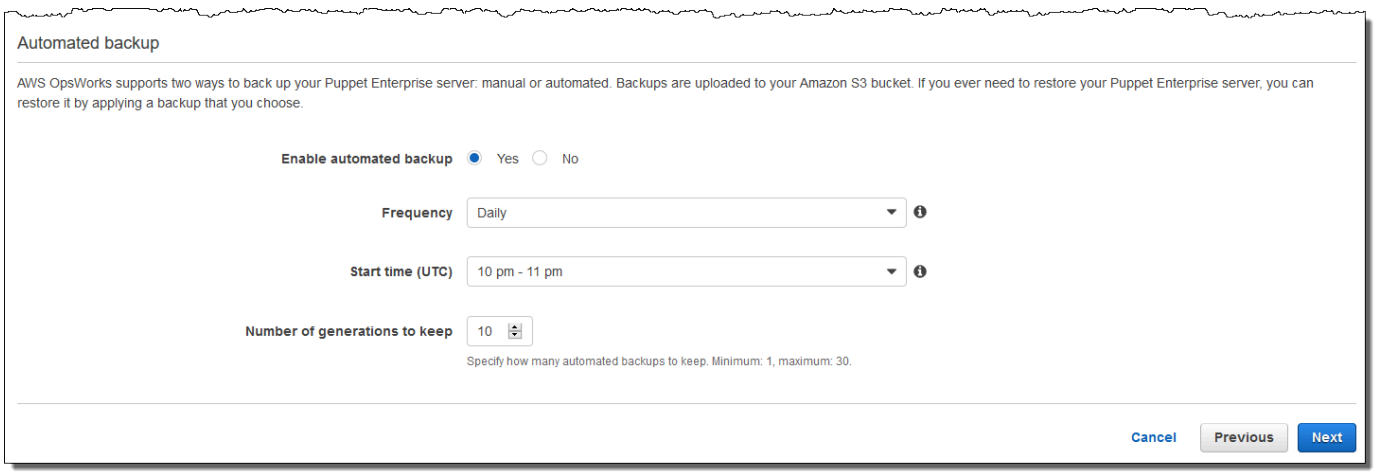


9. Dans la section System maintenance (Maintenance du système), définissez le jour et l'heure de début de la maintenance du système. Etant donné que le serveur doit être hors ligne lors de la maintenance du système, choisissez une heure où le serveur est peu sollicité pendant les heures normales de bureau.

   La fenêtre de maintenance est obligatoire. Vous pouvez modifier le jour et l'heure de début ultérieurement en utilisant le AWS Management Console AWS CLI, ou le APIs.



10. Configurez des sauvegardes. Par défaut, les sauvegardes automatiques sont activées. Définissez une fréquence et une heure préférées pour le démarrage de la sauvegarde automatique, et définissez le nombre de générations de sauvegarde à stocker dans Amazon Simple Storage Service. Un maximum de 30 sauvegardes peuvent être conservées ; lorsque le maximum est atteint, OpsWorks for Puppet Enterprise supprime les sauvegardes les plus anciennes pour faire de la place aux nouvelles.



11. (Facultatif) Dans Tags, ajoutez des balises au serveur et aux ressources associées, telles que l' EC2 instance, l'adresse IP élastique, le groupe de sécurité, le compartiment S3 et les sauvegardes. Pour plus d'informations sur le balisage d'un serveur OpsWorks pour Puppet Enterprise, consultez. Utilisation des balises sur les AWS OpsWorks for Puppet Enterprise ressources
12. Lorsque vous avez terminé de configurer les paramètres avancés, cliquez sur Suivant.
13. Sur la page Vérification, vérifiez vos choix. Lorsque vous êtes prêt à créer le serveur, choisissez Lancer.

    Pendant que vous attendez de AWS OpsWorks créer votre Puppet master, téléchargez le kit de démarrage Configurer le serveur maître Puppet à l'aide du Starter Kit et les informations d'identification de la console Puppet Enterprise. N'attendez pas que votre serveur soit en ligne pour télécharger ces éléments.

    Lorsque la création du serveur est terminée, votre Puppet Master est disponible sur la OpsWorks page d'accueil de Puppet Enterprise, avec le statut en ligne. Une fois que le serveur est en ligne, la console Puppet Enterprise est disponible sur le domaine du serveur, à une URL au format suivant : `https://`*`your_server_name`*`-`*`randomID`*`.`*`region`*`.opsworks-cm.io`.

Créez un Puppet Enterprise Master à l'aide du AWS CLI

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que

possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

La création d'un serveur principal OpsWorks pour Puppet Enterprise en exécutant des AWS CLI commandes est différente de la création d'un serveur dans la console. Dans la console, AWS OpsWorks crée un rôle de service et un groupe de sécurité pour vous, si vous ne spécifiez aucun rôle de service et un groupe de sécurité que vous souhaitez utiliser. Dans le AWS CLI, vous AWS OpsWorks pouvez créer un groupe de sécurité pour vous si vous n'en spécifiez pas un, mais cela ne crée pas automatiquement un rôle de service ; vous devez fournir un ARN de rôle de service dans le cadre de votre `create-server` commande. Dans la console, lors AWS OpsWorks de la création de votre Puppet master, vous téléchargez le kit de démarrage et les informations de connexion pour la console Puppet Enterprise. Comme vous ne pouvez pas le faire lorsque vous créez un master OpsWorks for Puppet Enterprise à l'aide de AWS CLI, vous utilisez un utilitaire de traitement JSON pour obtenir les informations d'identification et le kit de démarrage à partir des résultats de la `create-server` commande une fois que votre nouveau master OpsWorks for Puppet Enterprise est en ligne.

Si votre ordinateur local n'exécute pas déjà le AWS CLI, téléchargez-le et installez-le AWS CLI en suivant les instructions d'installation du guide de l'utilisateur de l'interface de ligne de commande AWS. Cette section ne décrit pas tous les paramètres que vous pouvez utiliser avec la commande `create-server`. Pour plus d'informations sur les paramètres `create-server`, consultez `create-server` dans la Référence de l'AWS CLI .

1. Veillez à exécuter les Prérequis. Pour créer votre serveur maître Puppet, vous avez besoin d'un ID de sous-réseau. Par conséquent, vous devez disposer d'un VPC.
2. Créez un rôle de service et un profil d'instance.
3. Recherchez et copiez les rôles ARNs de service dans votre compte.

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

   Dans les résultats de la commande `list-roles`, recherchez les entrées d'ARN de rôle de service ressemblant à ce qui suit. Notez le rôle du service ARNs. Ces valeurs vous seront utiles pour créer votre serveur maître Puppet Enterprise.

   ```
   {
       "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. Recherchez et copiez les profils ARNs d'instance de votre compte.

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

Dans les résultats de la commande `list-instance-profiles`, recherchez les entrées d'ARN de profil d'instance ressemblant à ce qui suit. Prenez note du profil de l'instance ARNs. Ces valeurs vous seront utiles pour créer votre serveur maître Puppet Enterprise.

```
{
    "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. Créez le master OpsWorks for Puppet Enterprise en exécutant la `create-server` commande.

   - La `--engine` valeur est `PuppetMonolithic`, `--engine-model` est et `--engine-version` peut être `2019` ou`2017`.
   - Le nom du serveur doit être unique dans votre AWS compte, dans chaque région. Les noms de serveur doivent commencer par une lettre. Ensuite, vous pouvez utiliser des lettres, des chiffres ou des tirets (-), jusqu'à un maximum de 40 caractères.

- Utilisez l'ARN du profil d'instance et celui du rôle de service que vous avez copiés lors des Étapes 3 et 4.
- Les types d'instance valides sont `m5.xlarge`, `c5.2xlarge` et `c5.4xlarge`. Pour plus d'informations sur les spécifications de ces types d'instances, consultez la section Types d'instances du guide de EC2 l'utilisateur Amazon.
- Le paramètre `--engine-attributes` est facultatif ; si vous ne spécifiez pas de mot de passe administrateur Puppet, le processus de création de serveur en génère un pour vous. Si vous ajoutez `--engine-attributes`, spécifiez un élément `PUPPET_ADMIN_PASSWORD`, un mot de passe administrateur permettant de se connecter à la page web de la console Puppet Enterprise. Le mot de passe doit comporter entre 8 et 32 caractères ASCII.
- Une paire de clés SSH est facultative, mais elle peut vous aider à vous connecter à votre serveur maître Puppet si vous devez réinitialiser le mot de passe de l'administrateur de la console. Pour plus d'informations sur la création d'une paire de clés SSH, consultez Amazon EC2 Key Pairs dans le guide de l' EC2 utilisateur Amazon.
- Pour utiliser un domaine personnalisé, ajoutez les paramètres suivants à votre commande. Sinon, le processus de création d'un maître Puppet génère automatiquement un point de terminaison pour vous. Les trois paramètres sont requis pour configurer un domaine personnalisé. Pour plus d'informations sur les exigences supplémentaires relatives à l'utilisation de ces paramètres, consultez CreateServerla référence de l'API AWS OpsWorks CM.
  - `--custom-domain` - Point de terminaison public facultatif d'un serveur, tel que `https://aws.my-company.com`.
  - `--custom-certificate` - Un certificat HTTPS au format PEM. La valeur peut être un seul certificat auto-signé ou une chaîne de certificats.
  - `--custom-private-key` - Une clé privée au format PEM pour se connecter au serveur à l'aide de HTTPS. La clé privée ne doit pas être chiffrée ; elle ne peut pas être protégée par un mot de passe ou une phrase passe.
- Une maintenance hebdomadaire du système est obligatoire. Vous devez spécifier des valeurs valides au format suivant : `DDD:HH:MM`. L'heure doit être exprimée en heure UTC (temps universel coordonné). Si vous ne spécifiez aucune valeur pour `--preferred-maintenance-window`, la valeur par défaut est un intervalle aléatoire d'une heure le mardi, le mercredi ou le vendredi.
- Les valeurs valides pour `--preferred-backup-window` doivent être spécifiées dans l'un des formats suivants : `HH:MM` pour les sauvegardes quotidiennes ou `DDD:HH:MM` pour les

sauvegardes hebdomadaires. L'heure indiquée est en heure UTC. La valeur par défaut est une heure de début quotidienne aléatoire. Pour annuler les sauvegardes quotidiennes, ajoutez le paramètre `--disable-automated-backup`.

- Pour`--security-group-ids`, entrez un ou plusieurs groupes de sécurité IDs, séparés par un espace.
- Pour `--subnet-ids`, entrez un ID de sous-réseau.

```
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
```

Voici un exemple.

```
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
```

L'exemple suivant crée un maître Puppet qui utilise un domaine personnalisé.

```
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
```

L'exemple suivant crée un maître Puppet qui ajoute deux balises : `Stage: Production` et `Department: Marketing`. Pour plus d'informations sur l'ajout et la gestion de balises sur OpsWorks les serveurs Puppet Enterprise, consultez Utilisation des balises sur les AWS OpsWorks for Puppet Enterprise ressources ce guide.

```
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 pour Puppet Enterprise, il faut environ 15 minutes pour créer un nouveau serveur. Ne supprimez pas la sortie de la commande `create-server` ou ne fermez pas votre session shell, car la sortie peut contenir des informations importantes qui ne seront plus réaffichées. Pour obtenir les mots de passe et le kit de démarrage à partir des résultats de la commande `create-server`, passez à l'étape suivante.

Si vous utilisez un domaine personnalisé avec le serveur, dans la sortie de la commande `create-server`, copiez la valeur de l'attribut `Endpoint`. Voici un exemple.

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

7. Si vous avez choisi que OpsWorks Puppet Enterprise génère un mot de passe pour vous, vous pouvez l'extraire dans un format utilisable à partir des `create-server` résultats en utilisant un processeur JSON tel que jq. Une fois que vous avez installé jq, vous pouvez exécuter les commandes suivantes pour extraire le mot de passe administrateur et le kit de démarrage Puppet. Si vous n'avez pas fourni votre propre mot de passe à l'étape 3, veillez à sauvegarder le mot de passe administrateur extrait dans un emplacement sécurisé, mais pratique.

```
#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
```

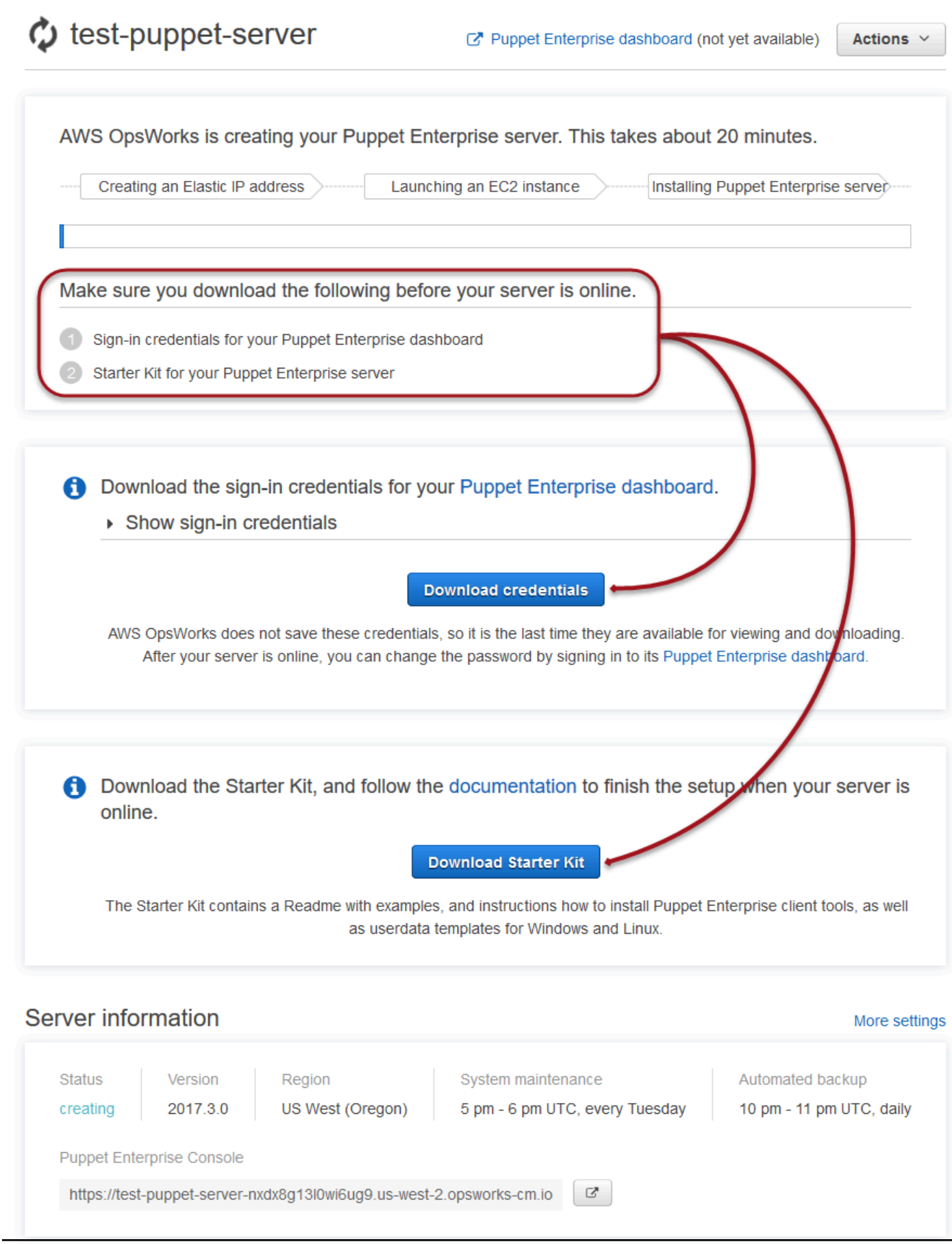
> **Note**
>
> Vous ne pouvez pas régénérer un nouveau kit de démarrage de serveur maître Puppet Starter Kit dans l' AWS Management Console. Lorsque vous créez un Puppet master à l'aide de AWS CLI, exécutez la `jq` commande précédente pour enregistrer le kit de démarrage codé en base64 dans les `create-server` résultats sous forme de fichier ZIP.

8. Si vous n'utilisez pas un domaine personnalisé, passez à l'étape suivante. Si vous utilisez un domaine personnalisé avec le serveur, créez une entrée CNAME dans l'outil de gestion DNS de votre entreprise pour faire pointer votre domaine personnalisé vers le point de terminaison OpsWorks pour Puppet Enterprise que vous avez copié à l'étape 6. Vous ne pouvez pas accéder ou vous connecter à un serveur avec un domaine personnalisé tant que vous n'avez pas effectué cette étape.
9. Passez à la section suivante, the section called “Finaliser la configuration”.

Configurer le serveur maître Puppet à l'aide du Starter Kit

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Alors que la création du Puppet Master est toujours en cours, la page Propriétés du serveur s'ouvre dans la console OpsWorks for Puppet Enterprise. La première fois que vous utilisez un nouveau serveur maître Puppet, la page de propriétés vous invite à télécharger deux éléments requis. Téléchargez ces éléments avant que votre serveur Puppet soit en ligne ; après, les boutons de téléchargement ne sont plus disponibles.


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

- Informations d'identification de connexion du serveur maître Puppet. Vous utiliserez ces informations d'identification pour vous connecter à la console Puppet Enterprise, où vous gérez la plupart des nœuds. AWS OpsWorks n'enregistre pas ces informations d'identification ; c'est la dernière fois qu'elles sont disponibles pour consultation et téléchargement. Si nécessaire, vous pouvez modifier le mot de passe qui est fourni avec ces informations d'identification une fois connecté.
- Starter Kit. Le Starter Kit contient un fichier README dans lequel figurent des informations et des exemples expliquant comment finaliser la configuration, ainsi que des informations d'identification d'administrateur pour la console Puppet Enterprise. De nouvelles informations d'identification sont générées (et les anciennes deviennent obsolètes) chaque fois que vous téléchargez le Starter Kit.

Prérequis

1. Alors que la création du serveur Puppet est encore en cours, téléchargez les informations d'identification de connexion pour le serveur Puppet, et sauvegardez-les dans un emplacement sécurisé, mais pratique.
2. Téléchargez le Starter Kit et décompressez le fichier .zip Starter Kit dans votre répertoire d'espace de travail. Ne partagez jamais vos informations d'identification de connexion avec qui que ce soit. Si d'autres utilisateurs doivent gérer le serveur maître Puppet, ajoutez-les ultérieurement comme administrateurs dans la console Puppet Enterprise. Pour plus d'informations sur la façon d'ajouter des utilisateurs au serveur maître Puppet, consultez Creating and managing users and user roles (Création et gestion d'utilisateurs et de rôles utilisateur) dans la documentation de Puppet Enterprise.

Installer le certificat de serveur maître Puppet

Pour travailler avec votre serveur maître Puppet et y ajouter des nœuds à gérer, vous devez installer le certificat correspondant. Installez-le en exécutant la AWS CLI commande suivante. Vous ne pouvez pas effectuer cette tâche dans le 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
```

Générer un jeton à court terme

Pour utiliser l'API Puppet, vous devez créer un jeton à court terme pour vous-même. Cette étape n'est pas nécessaire pour pouvoir utiliser la console Puppet Enterprise. Pour générer le jeton, exécutez la commande ci-dessous.

La durée de vie par défaut du jeton est de cinq minutes, mais il est possible de modifier cette valeur.

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

> **Note**
>
> La durée de vie par défaut du jeton étant de cinq minutes, le paramètre `--lifetime` est ajouté dans l'exemple de commande précédent afin d'allonger la durée de vie du jeton. Vous pouvez définir la durée de vie du jeton sur une période pouvant aller jusqu'à 10 ans (`10y`). Pour plus d'informations sur la façon de modifier la durée de vie par défaut du jeton, consultez Change the token's default lifetime (Modification de la durée de vie par défaut du jeton) dans la documentation de Puppet Enterprise.

Exemple de configuration du kit de démarrage Apache

Après avoir téléchargé et décompressé le kit de démarrage, vous pouvez utiliser la branche d'exemple dans le `control-repo-example` dossier d'exemple inclus pour configurer un serveur Web Apache sur vos nœuds gérés.

Le Starter Kit comprend deux dossiers `control-repo` : `control-repo` et `control-repo-example`. Le `control-repo` dossier inclut une `production` branche inchangée par rapport à ce que vous pourriez voir dans le GitHub référentiel Puppet. Le `control-repo-example` dossier contient également une `production` branche qui inclut un exemple de code pour configurer un serveur Apache avec un site Web de test.

1. Transmettez la branche `control-repo-example` `production` à votre git remote (l'URL `r10k_remote` de votre serveur maître Puppet). Dans le répertoire racine de votre Starter Kit, exécutez ce qui suit en le *`r10kRemoteUrl`* remplaçant par votre `r10k_remote` URL.

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

```
git push origin production
```

Puppet Code Manager utilise les branches Git comme environnements. Tous les nœuds se trouvent par défaut dans l'environnement de production.

> **Important**
>
> Vous ne devez jamais transmettre les données à une branche `master`. La branche `master` est réservée au serveur maître Puppet.

2. Déployez le code dans la branche `control-repo-example` vers votre serveur maître Puppet. Le serveur maître Puppet peut ainsi télécharger votre code Puppet depuis votre référentiel Git (`r10k_remote`). Dans le répertoire racine de votre Starter Kit, exécutez la commande ci-dessous.

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

Pour plus d'informations sur la façon dont vous pouvez appliquer l'exemple de configuration Apache aux nœuds gérés que vous créez dans Amazon EC2, consultez Étape 2 : Création d'instances en utilisant un script d'association sans surveillance ce guide.

Ajouter des nœuds à gérer par le serveur maître Puppet

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Rubriques

- Exécution des appels d'API associateNode()
- Considérations relatives à l'ajout de nœuds sur site
- En savoir plus

La méthode recommandée pour ajouter des nœuds consiste à utiliser l' AWS OpsWorks `associateNode()`API. Le serveur maître Puppet Enterprise héberge un référentiel qui vous permet d'installer le logiciel agent Puppet sur les nœuds à gérer, que ces derniers soient des ordinateurs physiques ou des machines virtuelles sur site. Le logiciel Puppet Agent de certains systèmes d'exploitation est installé sur le OpsWorks serveur Puppet Enterprise dans le cadre du processus de lancement. Le tableau suivant indique les agents du système d'exploitation disponibles sur votre serveur OpsWorks pour Puppet Enterprise au lancement.

Agents du système d'exploitation préinstallés

Système d'exploitation pris en charge	Versions
Ubuntu	16,04, 18,04, 20,04
Red Hat Enterprise Linux (RHEL)	6, 7, 8
Windows	Éditions 64 bits de toutes les versions Windows prises en charge par Puppet

Vous pouvez ajouter `puppet-agent` à votre serveur pour d'autres systèmes d'exploitation. Sachez que l'opération de maintenance du système supprimera les agents ajoutés à votre serveur après le lancement. Bien que l'archivage se poursuive pour la plupart des nœuds attachés existants qui exécutent déjà l'agent supprimé, les nœuds exécutant des systèmes d'exploitation Debian peuvent cesser d'envoyer des rapports. Nous vous recommandons d'effectuer l'installation manuellement `puppet-agent` sur les nœuds qui exécutent des systèmes d'exploitation pour lesquels le logiciel agent n'est pas préinstallé sur votre serveur OpsWorks pour Puppet Enterprise. Pour obtenir des informations détaillées sur la façon de rendre `puppet-agent` disponible sur votre serveur pour les nœuds associés à d'autres systèmes d'exploitation, consultez Installing agents (Installation d'agents) dans la documentation de Puppet Enterprise.

Pour plus d'informations sur la façon d'associer automatiquement des nœuds à votre Puppet master en renseignant les données utilisateur de l' EC2 instance, consultez. Ajout automatique de nœuds dans OpsWorks Puppet Enterprise

Exécution des appels d'API **`associateNode()`**

Après avoir ajouté des nœuds par installation`puppet-agent`, les nœuds envoient des demandes de signature de certificat (CSRs) au serveur OpsWorks for Puppet Enterprise. Vous pouvez les consulter CSRs dans la console Puppet ; pour plus d'informations sur le nœud CSRs, consultez la section

Gestion des demandes de signature de certificats dans la documentation de Puppet Enterprise. Exécuter le nœud CSRs de traitement des appels de `associateNode()` l'API OpsWorks for Puppet Enterprise et associer le nœud à votre serveur. Voici un exemple d'utilisation de cet appel d'API AWS CLI pour associer un seul nœud. Vous aurez besoin de la demande de signature de certificat au format PEM envoyée par le nœud, que vous pourrez obtenir à partir de la console Puppet.

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

Pour plus d'informations sur la façon d'ajouter automatiquement des nœuds en utilisant `associateNode()`, consultez Ajout automatique de nœuds dans OpsWorks Puppet Enterprise.

Considérations relatives à l'ajout de nœuds sur site

Après l'avoir installé `puppet-agent` sur vos ordinateurs locaux ou vos machines virtuelles, vous pouvez utiliser l'une des deux méthodes suivantes pour associer des nœuds locaux à votre master OpsWorks for Puppet Enterprise.

- Si un nœud prend en charge l'installation du kit SDK AWS, de l'AWS CLI ou de Outils AWS pour PowerShell, vous pouvez utiliser la méthode recommandée pour associer un nœud, qui consiste à exécuter un appel d'API `associateNode()`. Le kit de démarrage que vous téléchargez lorsque vous créez un master OpsWorks pour Puppet Enterprise pour la première fois montre comment attribuer des rôles aux nœuds à l'aide de balises. Vous pouvez appliquer des balises en même temps que vous associez des nœuds au serveur maître Puppet en spécifiant des faits approuvés dans la CSR. Par exemple, le référentiel de contrôle des démonstrations inclus dans le kit de démarrage est configuré pour utiliser la balise `pp_role` afin d'attribuer des rôles aux EC2 instances Amazon. Pour plus d'informations sur l'ajout de balises à la CSR en tant que faits approuvés, consultez Extension requests (permanent certificate data) dans la documentation de la plateforme Puppet.
- Si le nœud ne peut pas exécuter les outils de AWS gestion ou de développement, vous pouvez toujours l'enregistrer auprès de votre master OpsWorks for Puppet Enterprise de la même manière que vous l'enregistreriez auprès d'un master Puppet Enterprise non géré. Comme indiqué dans cette rubrique, l'installation `puppet-agent` envoie un CSR au maître OpsWorks de Puppet Enterprise. Un utilisateur autorisé de Puppet peut signer le CSR manuellement ou configurer la signature automatique CSRs en modifiant le `autosign.conf` fichier stocké sur le Puppet master. Pour plus d'informations sur la configuration de la signature automatique et la modification

de `autosign.conf`, consultez SSL configuration: autosigning certificate requests dans la documentation de la plateforme Puppet.

Pour associer des nœuds locaux à un Puppet master et autoriser ce dernier à tout accepter CSRs, procédez comme suit dans la console Puppet Enterprise. Le paramètre qui contrôle le comportement est `puppet_enterprise::profile::master::allow_unauthenticated_ca`.

> **Important**
>
> Pour des raisons de sécurité, il n' CSRs est pas recommandé d'activer le Puppet Master pour qu'il accepte l'auto-signature CSRs ou tout le monde. Par défaut, autoriser l'authentification non authentifiée CSRs rend un Puppet master accessible au monde entier. Le fait de définir le chargement des demandes de certificat de telle sorte qu'il soit activé par défaut peut rendre votre serveur maître Puppet vulnérable aux attaques de type déni de service (DoS, Denial of Service).

1. Connectez-vous à la console Puppet Enterprise.
2. Choisissez Configure, Classification, PE Master, puis l'onglet Configuration.
3. Dans l'onglet Classification, recherchez la classe puppet_enterprise::profile::master.
4. Définissez la valeur du paramètre allow_unauthenticated_ca sur true.
5. Enregistrez vos modifications. Vos modifications seront appliquées lors de la prochaine exécution de Puppet. Vous pouvez patienter 30 minutes que ces modifications prennent effet (et que les nœuds sur site soient ajoutés), ou vous pouvez lancer manuellement une exécution de Puppet dans la section Run de la console PE.

En savoir plus

Consultez le didacticiel Learn Puppet pour en savoir plus sur l'utilisation OpsWorks des serveurs Puppet Enterprise et sur les fonctionnalités de la console Puppet Enterprise.

Se connecter à la console Puppet Enterprise

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons

vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une fois que vous avez téléchargé les informations d'identification de connexion depuis la page de propriétés du serveur maître Puppet et que le serveur est en ligne, connectez-vous à la console Puppet Enterprise. Dans cette procédure, nous vous avons demandé d'indiquer votre référentiel de contrôle contenant vos modules et d'y ajouter au moins un nœud à gérer. Vous pouvez ainsi consulter les informations sur l'agent et les nœuds dans la console.

Lorsque vous essayez de vous connecter à la page Web de la console Puppet Enterprise, des avertissements relatifs aux certificats apparaissent dans votre navigateur jusqu'à ce que vous installiez un certificat SSL signé par une autorité de certification AWS OpsWorks spécifique sur l'ordinateur client que vous utilisez pour gérer votre serveur Puppet. Si vous préférez ne pas voir les avertissements avant de continuer vers la page web du tableau de bord, installez le certificat SSL avant de vous connecter.

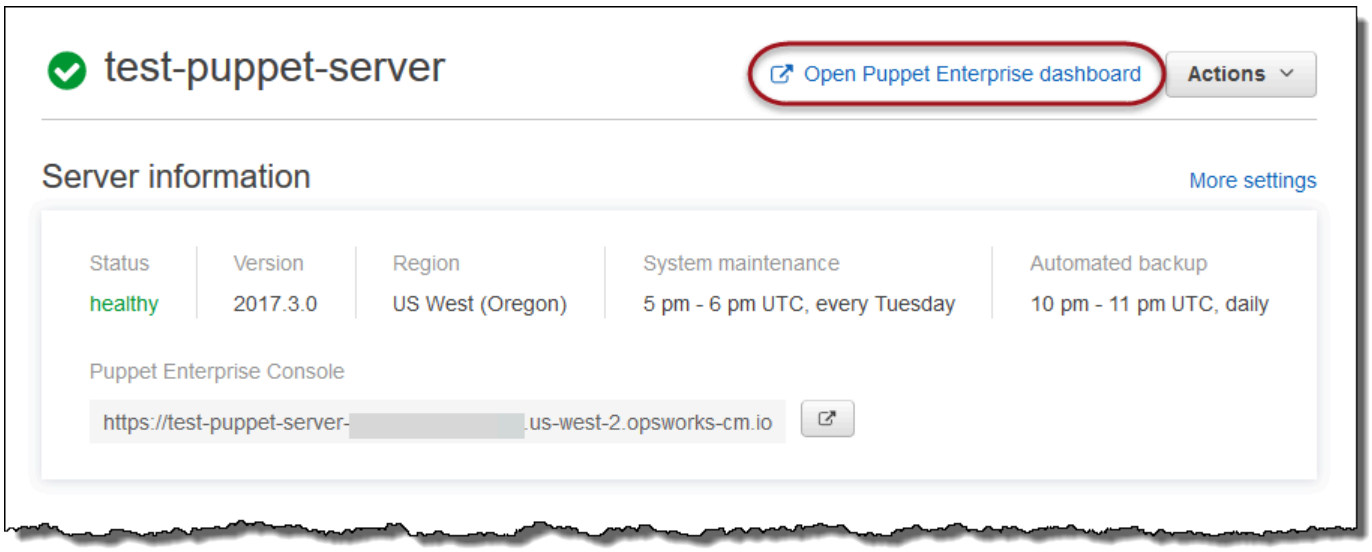
Pour installer le certificat AWS OpsWorks SSL

- Choisissez le certificat qui correspond à votre système.

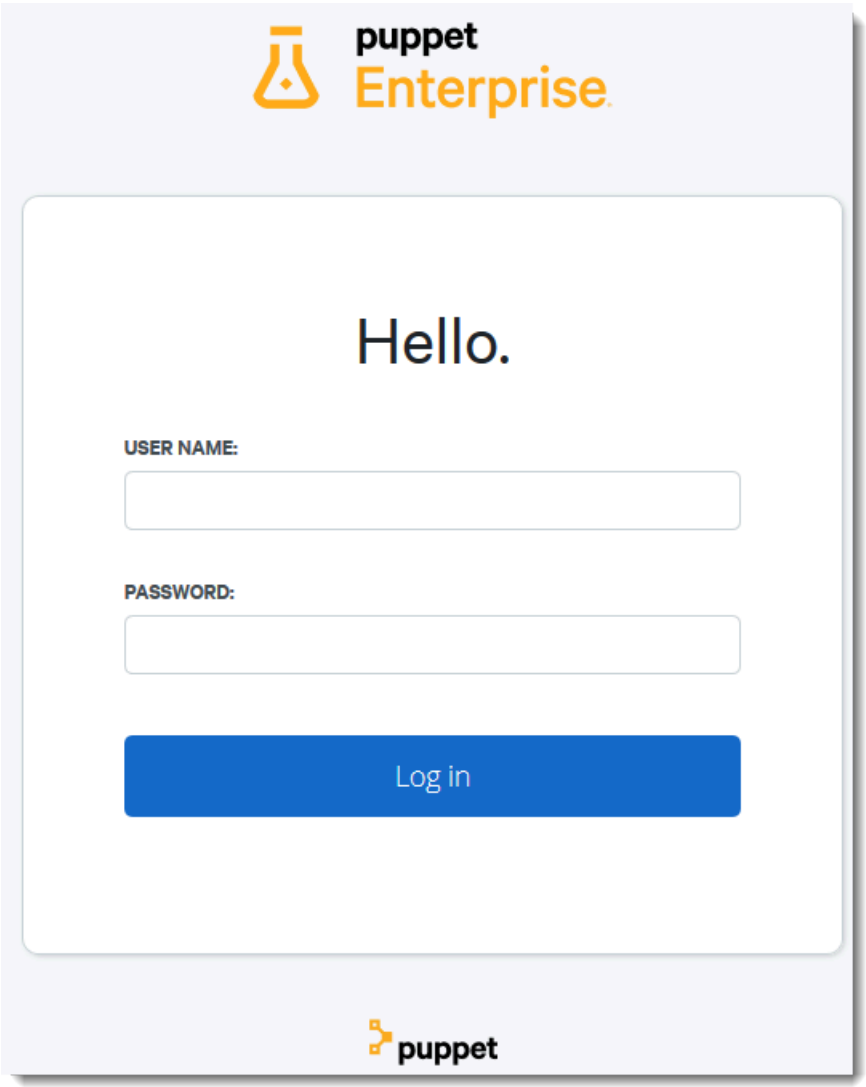
Après avoir installé le certificat SSL côté client, vous pouvez vous connecter à la console Puppet Enterprise sans voir de messages d'avertissement.

Pour se connecter à la console Puppet Enterprise

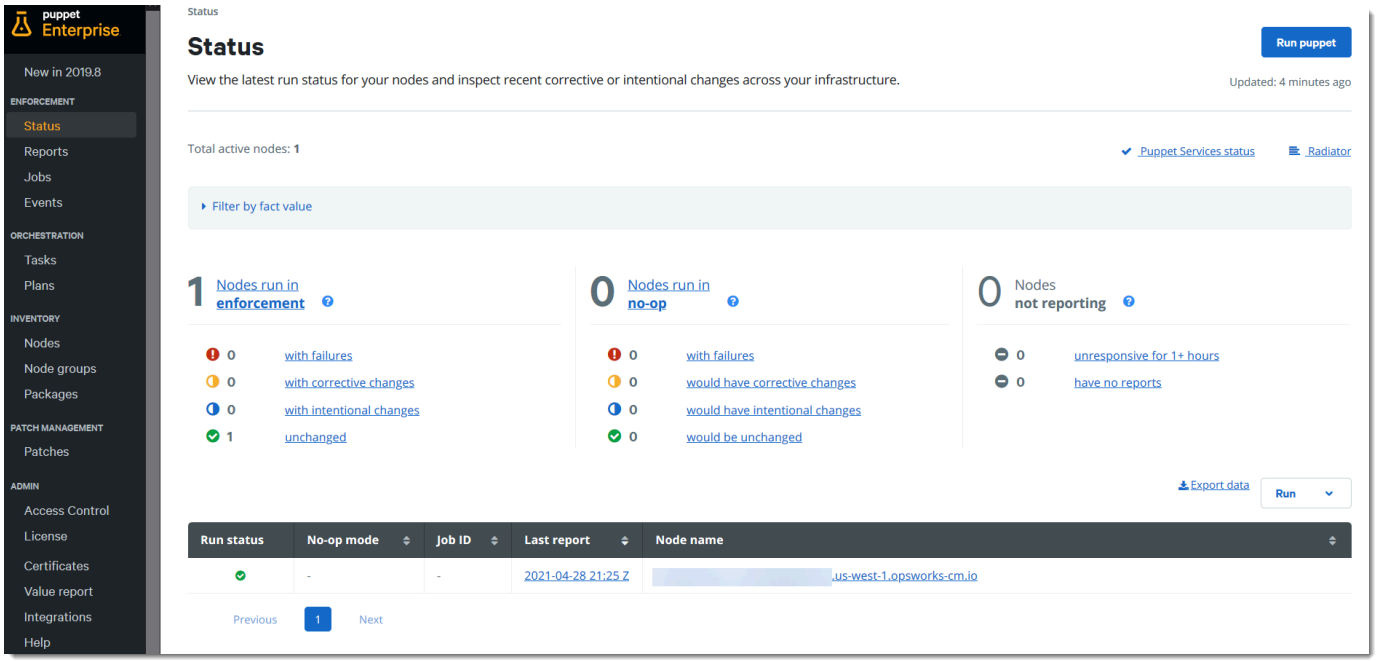
1. Décompressez et ouvrez les informations d'identification Puppet Enterprise que vous avez téléchargées à l'étape Prérequis. Vous aurez besoin de ces informations d'identification pour vous connecter.
2. Dans le AWS Management Console, ouvrez la page Propriétés de votre serveur Puppet.
3. En haut à droite de la page Properties, choisissez Open Puppet Enterprise console.



4. Connectez-vous à l'aide des informations d'identification de l'étape 1.



5. Dans la console Puppet Enterprise, vous pouvez voir les détails des nœuds que vous gérez, la progression de l'exécution du module et les événements, le niveau de conformité des nœuds, et plus encore. Pour plus d'informations sur les fonctionnalités de la console Puppet Enterprise et sur leur utilisation, consultez la section Gestion des nœuds dans la documentation de Puppet Enterprise.



Regrouper et classer des nœuds

Avant d'indiquer la configuration souhaitée pour vos nœuds en y appliquant des classes, regroupez-les en fonction de leur rôle au sein de votre entreprise ou selon leurs caractéristiques communes. Le fait de regrouper et de classer des nœuds implique les tâches de haut niveau ci-dessous. Vous pouvez effectuer ces tâches à l'aide de la console PE. Pour obtenir des informations détaillées sur la façon de regrouper et de classer vos nœuds, consultez Grouping and classifying nodes (Regroupement et classification de nœuds) dans la documentation de Puppet Enterprise.

1. Créez des groupes de nœuds.
2. Ajoutez des nœuds à ces groupes, manuellement ou automatiquement, en appliquant des règles que vous créez.
3. Attribuez des classes aux groupes de nœuds.

Réinitialiser les mots de passe administrateur et utilisateur

Pour plus d'informations sur la modification du mot de passe que vous utilisez pour vous connecter à la console Puppet Enterprise, voir Réinitialiser le mot de passe administrateur de la console dans la documentation de Puppet Enterprise.

Par défaut, l'accès à la console Puppet est verrouillé pour l'utilisateur après dix tentatives de connexion. Pour plus d'informations sur la façon de réinitialiser les mots de passe utilisateur en cas de verrouillage de l'accès, consultez Password endpoints (Points de terminaison des mots de passe) dans la documentation de Puppet Enterprise.

Facultatif : utilisation AWS CodeCommit en tant que référentiel de contrôle à distance Puppet r10k

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

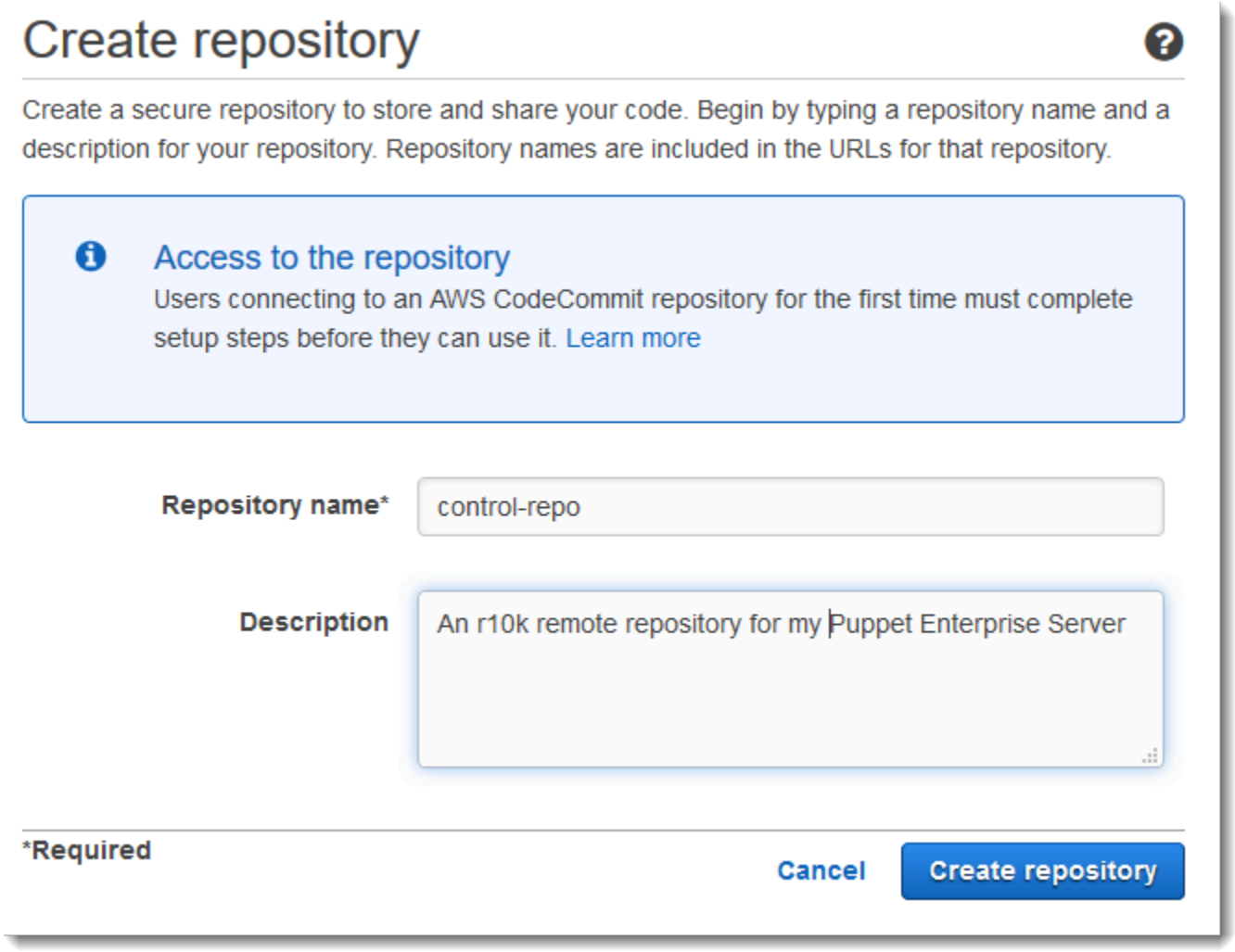
Vous pouvez créer un nouveau référentiel en utilisant et AWS CodeCommit en tant que référentiel de contrôle à distance r10k. Pour effectuer les étapes décrites dans cette section et utiliser un CodeCommit référentiel, vous avez besoin d'un utilisateur disposant des autorisations fournies par la politique AWSCodeCommitReadOnlygérée.

Rubriques

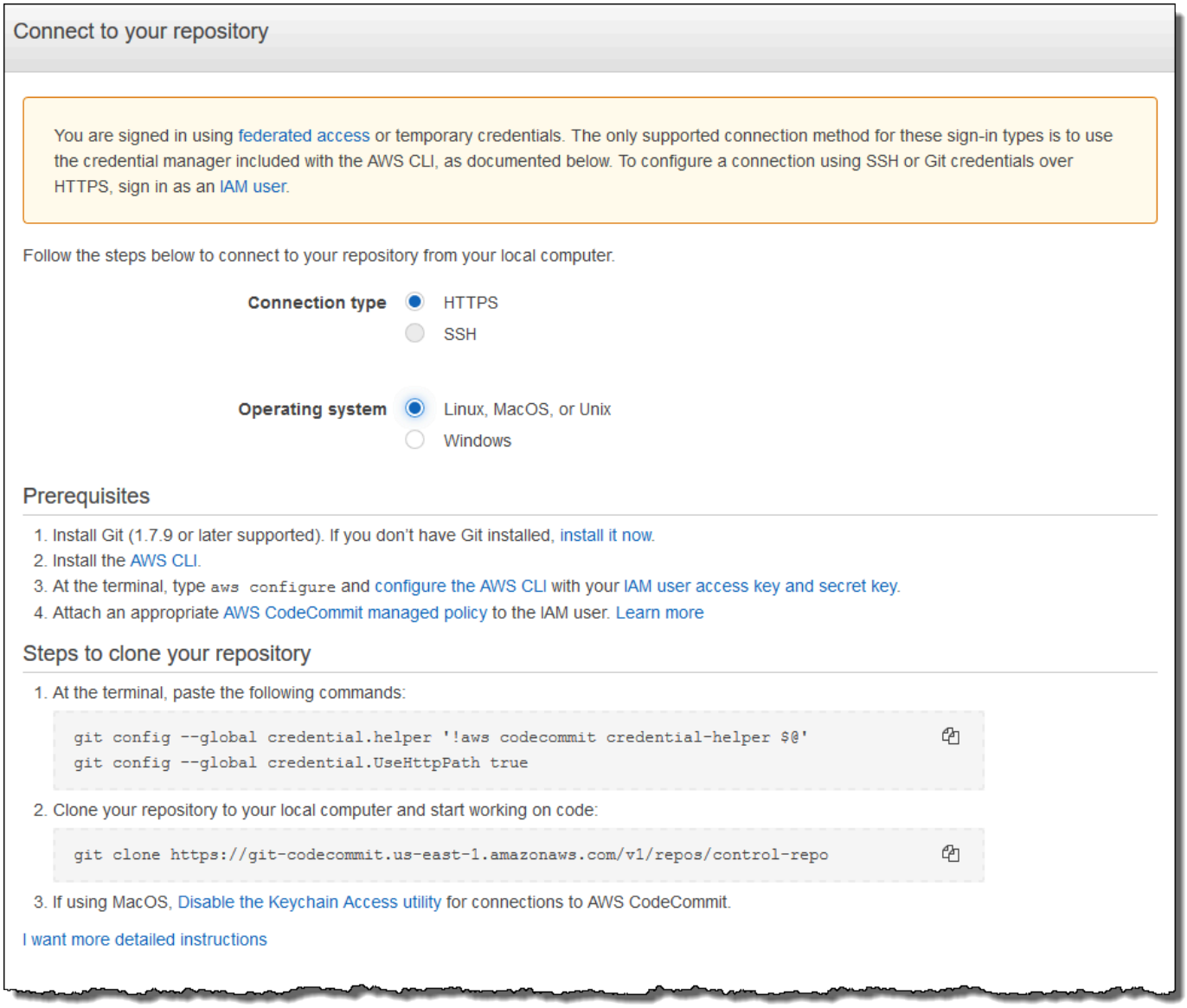
- Étape 1 : utilisation CodeCommit en tant que référentiel avec un type de connexion HTTPS
- Étape 2 : (Facultatif) Utilisation CodeCommit en tant que référentiel avec un type de connexion SSH

Étape 1 : utilisation CodeCommit en tant que référentiel avec un type de connexion HTTPS

1. Dans la CodeCommit console, créez un nouveau référentiel.



2. Choisissez Ignorer pour ne pas configurer de rubrique Amazon SNS.
3. Sur la page Code, choisissez Connectez-vous à votre référentiel.
4. Sur la page Connectez-vous à votre référentiel, choisissez HTTPS comme Type de connexion et choisissez votre système d'exploitation.



Dans la zoneÉtapes de clonage de votre référentiel, votre URL `git clone` doit se présenter sous la forme suivante : `https://git-codecommit.`*`region`*`.amazonaws.com/v1/repos/control-repo`. Copiez cette URL dans un endroit pratique, où vous pourrez facilement la retrouver pour configurer le serveur Puppet.

5. Fermez la page Connect to your repository, puis revenez à la OpsWorks configuration du serveur Puppet Enterprise.
6. Collez l'URL que vous avez copiée à l'étape 4 dans la zone r10k remote, sur la page Configure credentials (Configuration des informations d'identification) de l'assistant de configuration du serveur maître Puppet. Laissez la zone r10k private key (Clé privée r10k) vide. Finalisez la création et le lancement de votre serveur maître Puppet.

7. Dans la console IAM, associez la AWSCodeCommitReadOnlypolitique au rôle de profil d'instance de votre Puppet Master. Pour plus d'informations sur la façon d'ajouter une politique d'autorisations à un rôle IAM, consultez la section Ajout d'autorisations d'identité IAM (console) dans le guide de l'utilisateur IAM.
8. Suivez les étapes décrites dans la section Configuration pour les utilisateurs HTTPS à l'aide des informations d'identification Git du guide de AWS CodeCommit l'utilisateur pour transférer votre `control-repo` contenu existant vers le nouveau CodeCommit référentiel.
9. Vous pouvez à présent poursuivre en suivant les instructions de la section the section called "Finaliser la configuration" et utiliser le Starter Kit pour déployer le code sur votre serveur maître Puppet. Voici un exemple de commande.

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

Étape 2 : (Facultatif) Utilisation CodeCommit en tant que référentiel avec un type de connexion SSH

Vous pouvez configurer un référentiel de contrôle à distance AWS CodeCommit r10k pour utiliser l'authentification par paire de clés SSH. Les prérequis suivants doivent être remplis avant de démarrer cette procédure.

- Vous devez avoir lancé votre serveur OpsWorks pour Puppet Enterprise avec un référentiel de contrôle HTTPS, the section called "Étape 1 : utilisation CodeCommit en tant que référentiel avec un type de connexion HTTPS" comme décrit dans la section précédente. Ceci doit être effectué en premier afin de pouvoir charger la configuration requise dans le maître Puppet.
- Assurez-vous que la politique AWSCodeCommitReadOnlygérée est attachée à un utilisateur. Pour plus d'informations sur la création d'un utilisateur, consultez la section Création d'un utilisateur IAM dans votre AWS compte dans le guide de l'utilisateur IAM.
- Créez et associez une clé SSH avec votre utilisateur . Suivez les instructions pour créer une paire de public/private clés `ssh-keygen` à l'étape 3 : Configurer les informations d'identification sous Linux, macOS ou Unix dans le guide de l'AWS CodeCommit utilisateur.

1. Au AWS CLI cours d'une session, exécutez la commande suivante pour télécharger le contenu du fichier de clé privée dans AWS Systems Manager Parameter Store. Votre serveur OpsWorks for Puppet Enterprise interroge ce paramètre pour obtenir le fichier de certificat requis. *`private_key_file`*Remplacez-le par le chemin d'accès à votre fichier de clé privée SSH.

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

2. Ajoutez les autorisations du magasin de paramètres de Systems Manager à votre Puppet Master.

   a. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.

   b. Dans le volet de navigation de gauche, choisissez Rôles.

   c. Choisissez aws-opsworks-cm-ec2 rôles.

   d. Sous l'onglet Permissions (Autorisations), choisissez Attach policies (Attacher des politiques).

   e. Dans la barre Rechercher, saisissez **AmazonSSMManagedInstanceCore**.

   f. Dans les résultats de recherche, sélectionnez Amazon SSMManaged InstanceCore.

   g. Choisissez Attach policy (Attacher la politique).

3. Créez le fichier manifeste de configuration. Si vous utilisez le référentiel `control-repo-example` fourni dans le kit de démarrage, créez les fichiers suivants aux emplacements indiqués dans l'exemple de référentiel. Vous pouvez également les créer conformément à la structure de votre propre référentiel de contrôle. Remplacez la *IAM_USER_SSH_KEY* valeur par l'ID de clé SSH que vous avez créé dans les conditions préalables de cette procédure.

```
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. Transférez votre référentiel de contrôle vers CodeCommit. Exécutez les commandes suivantes pour transférer le nouveau fichier manifeste vers votre référentiel.

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

5. Déployez les fichiers manifestes. Exécutez les commandes suivantes pour déployer la configuration mise à jour sur votre serveur OpsWorks for Puppet Enterprise. Remplacez *STARTER_KIT_DIRECTORY* par le chemin d'accès à vos fichiers de configuration Puppet.

```
cd STARTER_KIT_DIRECTORY

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

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

6. Mettez à jour la classification du serveur OpsWorks for Puppet Enterprise. Par défaut, l'agent Puppet s'exécute sur les nœuds (y compris le maître) toutes les 30 minutes. Pour éviter les temps d'attente, vous pouvez exécuter manuellement l'agent sur le maître Puppet. L'exécution de l'agent récupère le nouveau fichier manifeste.

   a. Connectez-vous à la console Puppet Enterprise.

   b. Choisissez Classification.

   c. Développez PE Infrastructure.

   d. Choisissez PE Master.

   e. Dans l'onglet Configuration, saisissez **`profile::codecommit`** dans Add new class (Ajouter une nouvelle classe).

      La nouvelle classe, `profile::codecommit`, peut ne pas s'afficher immédiatement après l'exécution de `puppet-code deploy`. Choisissez Actualiser sur cette page si elle n'apparaît pas.

   f. Choisissez Add class (Ajouter une classe), puis choisissez Commit 1 change (Valider 1 changement).

   g. Exécutez manuellement l'agent Puppet sur le serveur OpsWorks for Puppet Enterprise. Choisissez Nodes (Nœuds), sélectionnez votre serveur dans la liste, choisissez Run Puppet (Exécuter Puppet), puis choisissez Run (Exécuter).

7. Dans la console Puppet Enterprise, remplacez l'URL du référentiel pour utiliser le protocole SSH plutôt que HTTPS. La configuration que vous effectuez au cours de ces étapes est enregistrée OpsWorks pendant le processus de sauvegarde et de restauration de Puppet Enterprise. Il n'est donc pas nécessaire de modifier manuellement la configuration du référentiel après les activités de maintenance.

   a. Choisissez Classification.

   b. Développez PE Infrastructure.

   c. Choisissez PE Master.

   d. Dans l'onglet Configuration, trouvez la classe `puppet_enterprise::profile::master`.

   e. Choisissez Modifier en regard du paramètre `r10k_remote`.

   f. Remplacez l'URL HTTPS par l'URL SSH pour votre référentiel, puis choisissez Commit 1 change (Valider 1 changement).

   g. Exécutez manuellement l'agent Puppet sur le serveur OpsWorks for Puppet Enterprise. Choisissez Nodes (Nœuds), sélectionnez votre serveur dans la liste, choisissez Run Puppet (Exécuter Puppet), puis choisissez Run (Exécuter).

Créez un AWS OpsWorks for Puppet Enterprise Master en utilisant AWS CloudFormation

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks for Puppet Enterprise vous permet d'exécuter un serveur Puppet Enterprise dans AWS. Il ne faut que 15 minutes environ pour mettre en service un serveur maître Puppet Enterprise.

À compter du 3 mai 2021, OpsWorks for Puppet Enterprise stocke certains attributs du serveur Puppet Enterprise dans. AWS Secrets Manager Pour de plus amples informations, veuillez consulter Intégration avec AWS Secrets Manager.

La procédure pas à pas suivante vous aide à créer un Puppet Master dans OpsWorks Puppet Enterprise en créant un stack in. AWS CloudFormation

Rubriques

- Prérequis
- Créer un serveur maître Puppet Enterprise dans AWS CloudFormation

Prérequis

Avant de créer un nouveau Puppet master, créez les ressources externes à Puppet Enterprise dont vous aurez besoin OpsWorks pour accéder à votre Puppet master et le gérer. Pour plus d'informations, consultez Prérequis dans la section Mise en route de ce guide.

Si vous créez un serveur qui utilise un domaine personnalisé, vous avez besoin d'un domaine, d'un certificat et d'une clé privée personnalisés. Vous devez spécifier des valeurs pour ces trois paramètres dans votre AWS CloudFormation modèle. Pour plus d'informations sur les exigences relatives aux `CustomPrivateKey` paramètres`CustomDomain`,, et`CustomCertificate`,,,,,,, consultez CreateServerle manuel de référence de l'API AWS OpsWorks CM.

Consultez la section OpsWorks -CM de la référence du modèle de guide de AWS CloudFormation l'utilisateur pour en savoir plus sur les valeurs prises en charge et requises dans le AWS CloudFormation modèle que vous utilisez pour créer votre serveur.

Créer un serveur maître Puppet Enterprise dans AWS CloudFormation

Cette section décrit comment utiliser un AWS CloudFormation modèle pour créer une pile qui crée un serveur principal OpsWorks pour Puppet Enterprise. Vous pouvez le faire à l'aide de la AWS CloudFormation console ou du AWS CLI. Un exemple de AWS CloudFormation modèle est à votre disposition pour créer une pile de serveurs OpsWorks pour Puppet Enterprise. Veillez à mettre à jour l'exemple de modèle avec votre propre nom de serveur, vos rôles IAM, votre profil d'instance, la description du serveur, le nombre de sauvegardes conservées, les options de maintenance et les balises facultatives. Si votre serveur utilise un domaine personnalisé, vous devez spécifier des valeurs pour les paramètres `CustomDomain`, `CustomCertificate`, et `CustomPrivateKey` dans votre modèle AWS CloudFormation . Pour plus d'informations sur ces options, consultez the section called “Créez un Puppet Enterprise Master à l'aide du AWS Management Console” dans la section Mise en route de ce guide.

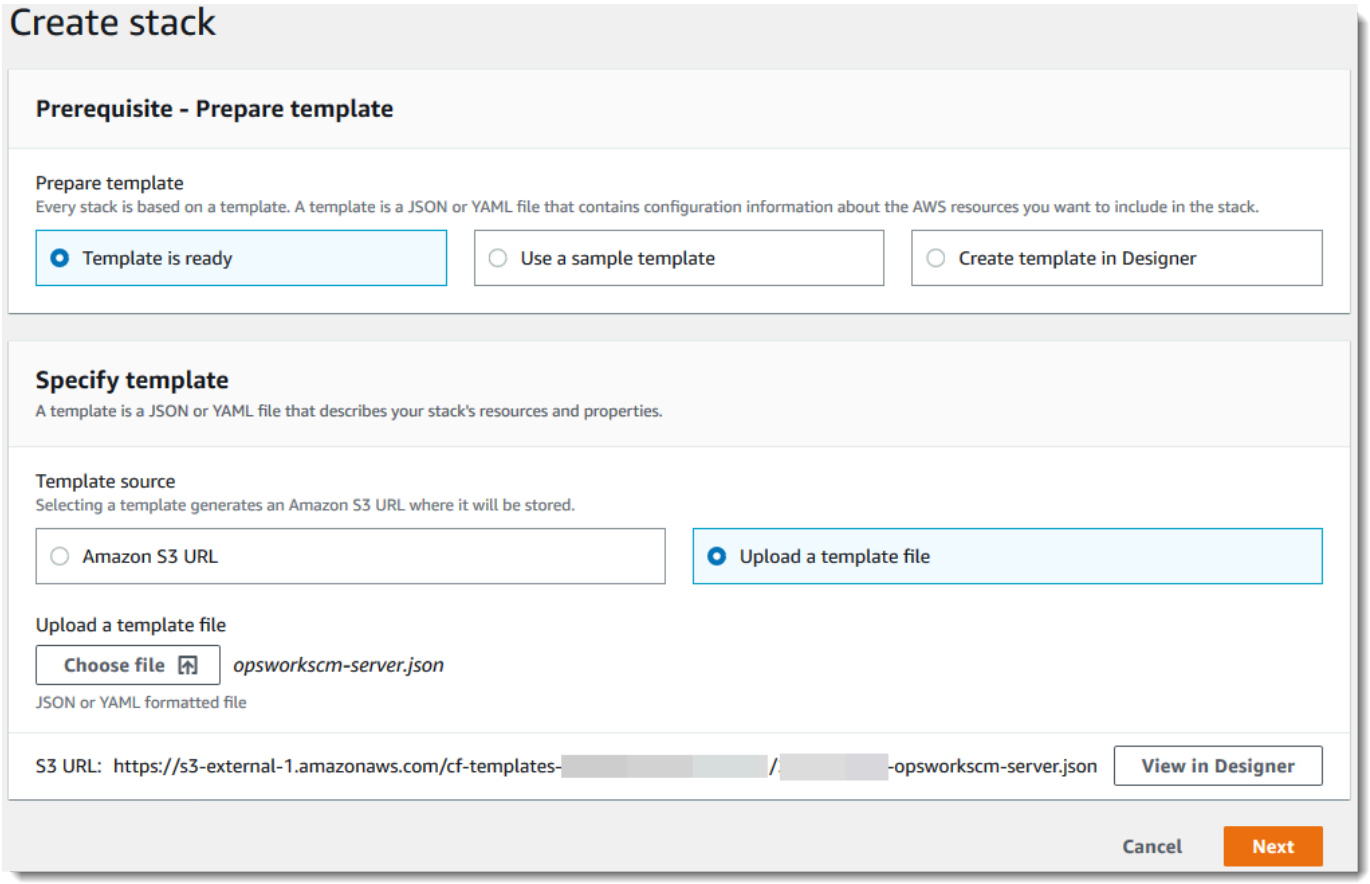
Rubriques

- Créez un Puppet Enterprise Master à l'aide de AWS CloudFormation (Console)
- Créez un Puppet Enterprise Master à l'aide de AWS CloudFormation (CLI)

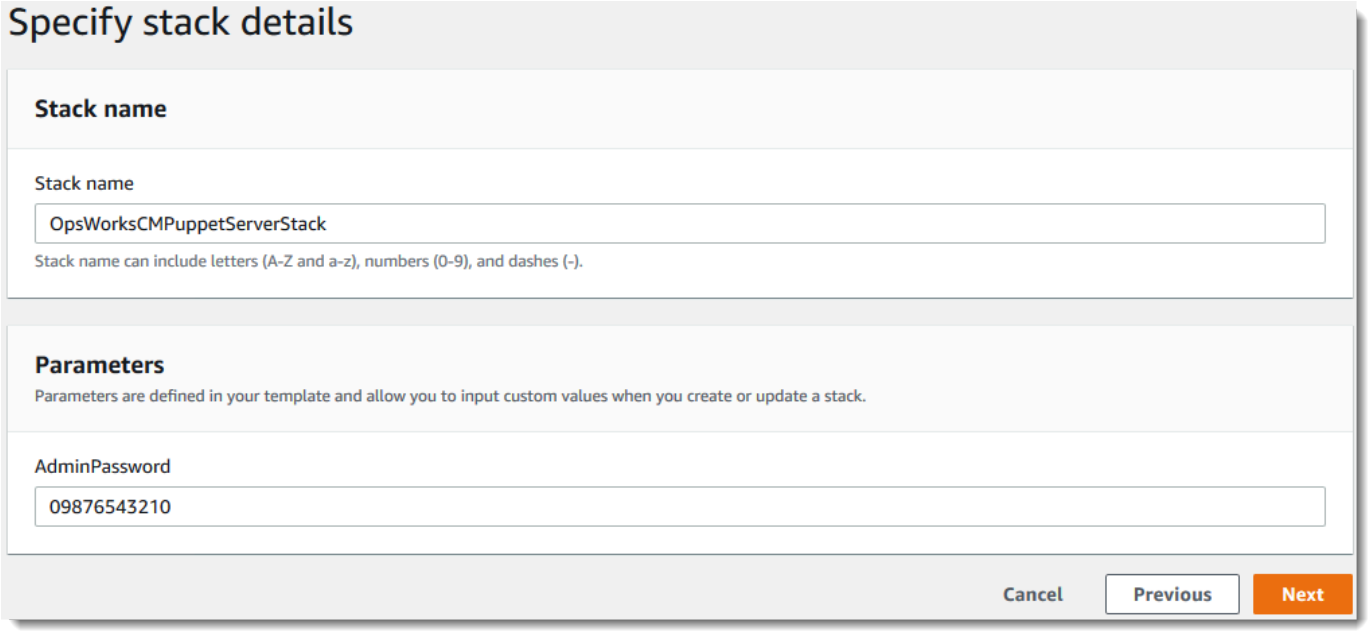
Créez un Puppet Enterprise Master à l'aide de AWS CloudFormation (Console)

1. Connectez-vous à la AWS CloudFormation console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudformation.
2. Sur la page d' AWS CloudFormation accueil, choisissez Create stack.
3. Dans Prerequisite - Prepare template (Prérequis - Préparation du modèle), si vous utilisez l'exemple de modèle AWS CloudFormation, choisissez Template is ready (Le modèle est prêt).
4. Dans Specify template (Spécifier le modèle), choisissez la source de votre modèle. Pour cette procédure pas à pas, choisissez Télécharger un fichier modèle, puis téléchargez un AWS CloudFormation modèle qui crée un serveur Puppet Enterprise. Accédez à votre fichier de modèle, puis choisissez Suivant.

   Un AWS CloudFormation modèle peut être au format YAML ou JSON. Un AWS CloudFormation modèle d'exemple est à votre disposition ; veillez à remplacer les valeurs d'exemple par les vôtres. Vous pouvez utiliser le concepteur de AWS CloudFormation modèles pour créer un nouveau modèle ou valider un modèle existant. Pour plus d'informations sur la procédure, consultez Présentation de l'interface de conception AWS CloudFormation dans le guide de l'utilisateur AWS CloudFormation .



5. Sur la page Specify stack details (Spécifier les détails de la pile), saisissez un nom pour votre pile. Il doit être différent du nom de votre serveur, car il s'agit uniquement d'un nom de pile. Dans la zone Paramètres, saisissez un mot de passe administrateur permettant de se connecter à la page web de la console Puppet Enterprise. Le mot de passe doit comporter entre 8 et 32 caractères ASCII. Choisissez Suivant.



6. Sur la page Options, vous pouvez ajouter des balises au serveur que vous créez avec la pile et choisir un rôle IAM pour créer des ressources si vous n'avez pas encore spécifié de rôle IAM à utiliser dans votre modèle. Une fois les options spécifiées, choisissez Suivant. Pour plus d'informations sur les options avancées telles que les déclencheurs d'annulation, consultez la section Configuration des options de AWS CloudFormation pile dans le guide de l'AWS CloudFormation utilisateur.
7. Sur la page Vérification, vérifiez vos choix. Lorsque vous êtes prêt à créer votre pile de serveur, choisissez Créer.

   Pendant que vous attendez de AWS CloudFormation créer la pile, consultez l'état de création de la pile. Si la création de la pile échoue, examinez les messages d'erreur affichés dans la console pour vous aider à résoudre le problème. Pour plus d'informations sur la résolution des erreurs dans les piles AWS CloudFormation , consultez Dépannage des erreurs dans le guide de l'utilisateur AWS CloudFormation .

   Lorsque la création du serveur est terminée, votre Puppet Master est disponible sur la OpsWorks page d'accueil de Puppet Enterprise, avec le statut en ligne. Une fois que le serveur est en ligne, la console Puppet Enterprise est disponible sur le domaine du serveur, à une URL au format suivant : `https://your_server_name-randomID.region.opsworks-cm.io`.

> **Note**
>
> Si vous avez spécifié un domaine, un certificat et une clé privée personnalisés pour votre serveur, créez une entrée CNAME dans l'outil de gestion DNS de votre entreprise qui mappe votre domaine personnalisé au point de terminaison généré automatiquement OpsWorks pour le serveur par Puppet Enterprise. Vous ne pouvez pas gérer le serveur ou vous connecter au site web de gestion Puppet Enterprise pour le serveur tant que vous n'avez pas mappé le point de terminaison généré à votre valeur de domaine personnalisée.
> Pour obtenir la valeur du point de terminaison généré, exécutez la AWS CLI commande suivante une fois que votre serveur est en ligne :
>
> ```
> aws opsworks describe-servers --server-name server_name
> ```

Créez un Puppet Enterprise Master à l'aide de AWS CloudFormation (CLI)

Si votre ordinateur local n'exécute pas déjà le AWS CLI, téléchargez-le et installez-le AWS CLI en suivant les instructions d'installation du guide de l'utilisateur de l'interface de ligne de commande AWS. Cette section ne décrit pas tous les paramètres que vous pouvez utiliser avec la commande create-stack. Pour plus d'informations sur les paramètres create-stack, consultez create-stack dans la Référence de l'AWS CLI .

1. Assurez-vous de compléter le formulaire de Prérequis création d'un master OpsWorks pour Puppet Enterprise.
2. Créez un rôle de service et un profil d'instance.

   Recherchez et copiez les rôles ARNs de service dans votre compte.

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

   Dans les résultats de la commande `list-roles`, recherchez les entrées de rôle de service et de profil d'instance ressemblant à ce qui suit. Notez le rôle de service et le profil ARNs d'instance, puis ajoutez-les au AWS CloudFormation modèle que vous utilisez pour créer votre pile de serveurs Puppet Master.

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

3. Créez le master OpsWorks for Puppet Enterprise en exécutant à nouveau la create-stack commande.

- *stack_name*Remplacez-le par le nom de votre pile. C'est le nom de la AWS CloudFormation pile, pas celui de votre Puppet Master. Le serveur maître Puppet correspond à la valeur de `ServerName` dans votre modèle AWS CloudFormation .
- *template*Remplacez-le par le chemin d'accès à votre fichier modèle et l'extension *yaml or json* par `.yaml` ou `.json` selon le cas.
- Les valeurs de `--parameters` correspondent à celles EngineAttributesde l'CreateServerAPI. Pour Puppet, les éléments suivants sont des attributs de moteur fournis par l'utilisateur pour créer un serveur. Les attributs de moteur r10k connectent votre serveur maître à un référentiel de code pour gérer la configuration d'environnement du serveur. Pour plus d'informations sur les attributs de moteur r10k, consultez Gestion de code avec r10k dans la documentation Puppet Enterprise.
  - `PUPPET_ADMIN_PASSWORD`, un mot de passe administrateur permettant de se connecter à la page web de la console Puppet Enterprise. Le mot de passe doit comporter entre 8 et 32 caractères ASCII, et inclure au moins une lettre majuscule, une lettre minuscule, un chiffre et un caractère spécial.
  - `PUPPET_R10K_REMOTE`, l'URL de votre référentiel de contrôle (par exemple, ssh://git@your.git-repo.com:user/control-repo.git). La spécification d'un référentiel de contrôle distant r10k ouvre un port TCP 8170.
  - `PUPPET_R10K_PRIVATE_KEY`. Si vous utilisez un dépôt Git privé, ajoutez PUPPET_R10K_PRIVATE_KEY pour spécifier une URL SSH et une clé SSH privée codée PEM.

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

Voici un exemple.

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

L'exemple suivant spécifie les attributs du moteur r10k en tant que paramètres, lorsqu'ils ne sont pas fournis dans le AWS CloudFormation modèle. Un exemple de modèle qui inclut les attributs

de moteur r10k, `puppet-server-param-attributes.yaml`, est inclus dans les exemples de modèle AWS CloudFormation.

```
aws cloudformation create-stack --stack-name MyPuppetStack --
template-body file://puppet-server-param-attributes.yaml --parameters
 ParameterKey=AdminPassword,ParameterValue="superSecret1%3"
 ParameterKey=R10KRemote,ParameterValue="https://www.yourRemote.com"
 ParameterKey=R10KKey,ParameterValue="$(cat puppet-r10k.pem)"
```

L'exemple suivant spécifie des attributs de moteur r10k et leurs valeurs dans le modèle AWS CloudFormation ; la commande doit uniquement pointer vers le fichier de modèle. Le modèle spécifié comme valeur de `--template-body`, `puppet-server-in-file-attributes.yaml`, est inclus dans les exemples de modèle AWS CloudFormation.

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

4. (Facultatif) Pour obtenir le statut de la création de la pile, exécutez la commande suivante.

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

5. Une fois la pile créée, passez à la section suivante, the section called “Finaliser la configuration”. Si la création de la pile échoue, examinez les messages d'erreur affichés dans la console pour vous aider à résoudre le problème. Pour plus d'informations sur la résolution des erreurs dans les AWS CloudFormation piles, consultez la section Résolution des erreurs dans le guide de l'AWS CloudFormation utilisateur.

Mettre à jour un serveur OpsWorks pour Puppet Enterprise afin d'utiliser un domaine personnalisé

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section décrit comment mettre à jour un serveur OpsWorks pour Puppet Enterprise existant afin d'utiliser un domaine et un certificat personnalisés en utilisant une sauvegarde du serveur pour créer un nouveau serveur. Essentiellement, vous copiez un serveur existant OpsWorks pour Puppet Enterprise 2.0 en créant un nouveau serveur à partir d'une sauvegarde, puis en configurant le nouveau serveur pour qu'il utilise un domaine, un certificat et une clé privée personnalisés.

Rubriques

- Prérequis
- Limites
- Mise à jour d'un serveur pour utiliser un domaine personnalisé
- consultez aussi

Prérequis

Vous trouverez ci-dessous les conditions requises pour mettre à jour un serveur Puppet Enterprise existant OpsWorks afin d'utiliser un domaine et un certificat personnalisés.

- Le serveur que vous souhaitez mettre à jour (ou copier) doit exécuter Puppet Enterprise 2019.8.5.
- Choisissez la sauvegarde à utiliser pour créer un nouveau serveur. Vous devez disposer d'au moins une sauvegarde du serveur que vous souhaitez mettre à jour. Pour plus d'informations sur les sauvegardes dans OpsWorks Puppet Enterprise, consultezSauvegardez et OpsWorks pour Puppet Enterprise Server.
- Préparez le rôle de service et le profil d'instance ARNs que vous avez utilisés pour créer le serveur existant qui est à l'origine de votre sauvegarde.
- Assurez-vous que vous exécutez la version la plus récente de l' AWS CLI. Pour plus d'informations sur la mise à jour de vos AWS CLI outils, consultez le guide de AWS CLI l'utilisateur de l'interface de ligne de commande AWS sur l'installation.

Limites

Lorsque vous mettez à jour un serveur existant en créant un nouveau serveur à partir d'une sauvegarde, le nouveau serveur ne peut pas être exactement le même que celui existant OpsWorks pour le serveur Puppet Enterprise.

- Vous ne pouvez effectuer cette procédure qu'en utilisant le AWS CLI ou l'un des AWS SDKs. Vous ne pouvez pas créer un nouveau serveur à partir d'une sauvegarde avec AWS Management Console.
- Le nouveau serveur ne peut pas utiliser le même nom que le serveur existant au sein d'un compte et d'une région AWS. Le nom doit être différent de celui du serveur existant que vous avez utilisé comme source de la sauvegarde.
- Les nœuds attachés au serveur existant ne sont pas gérés par le nouveau serveur. Vous devez effectuer l'une des opérations suivantes.
  - Attachez des nœuds différents, car des nœuds ne peuvent pas être gérés par plusieurs maîtres Puppet.
  - Migrez les nœuds à partir du serveur existant (source de la sauvegarde) vers le nouveau serveur et le nouveau point de terminaison de domaine personnalisé. Pour de plus amples informations sur la migration des nœuds, veuillez consulter la documentation Puppet Enterprise.

Mise à jour d'un serveur pour utiliser un domaine personnalisé

Pour mettre à jour un maître Puppet existant, vous en faites une copie en exécutant la commande `create-server`, et en ajoutant des paramètres pour spécifier une sauvegarde, un domaine personnalisé, un certificat personnalisé et une clé privée personnalisée.

1. Si aucun rôle de service ou profil d'instance n' ARNs est disponible à spécifier dans votre `create-server` commande, suivez les étapes 1 Créez un serveur Chef Automate à l'aide du AWS CLI à 5 pour créer un rôle de service ou un profil d'instance que vous pouvez utiliser.
2. Si vous ne l'avez pas déjà fait, recherchez la sauvegarde du maître Puppet existant sur lequel vous souhaitez baser un nouveau serveur avec un domaine personnalisé. Exécutez la commande suivante pour afficher des informations sur toutes les OpsWorks sauvegardes de Puppet Enterprise dans votre compte et dans une région. Veillez à notez l'ID de la sauvegarde que vous souhaitez utiliser.

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

3. Créez le serveur OpsWorks for Puppet Enterprise en exécutant la `create-server` commande.
   - La `--engine` valeur est `PuppetMonolithic`, `--engine-model` est et `--engine-version` est `2019` ou`2017`.

- Le nom du serveur doit être unique dans votre AWS compte, dans chaque région. Les noms de serveur doivent commencer par une lettre. Ensuite, vous pouvez utiliser des lettres, des chiffres ou des tirets (-), jusqu'à un maximum de 40 caractères.
- Utilisez l'ARN du profil d'instance et celui du rôle de service que vous avez copiés lors des Étapes 3 et 4.
- Les types d'instance valides sont `c4.large`, `c4.xlarge` et `c4.2xlarge`. Pour plus d'informations sur les spécifications de ces types d'instances, consultez la section Types d'instances du guide de EC2 l'utilisateur Amazon.
- Le paramètre `--engine-attributes` est facultatif ; si vous ne spécifiez pas de mot de passe administrateur Puppet, le processus de création de serveur en génère un pour vous. Si vous ajoutez `--engine-attributes`, spécifiez un élément `PUPPET_ADMIN_PASSWORD`, un mot de passe administrateur permettant de se connecter à la page web de la console Puppet Enterprise. Le mot de passe doit comporter entre 8 et 32 caractères ASCII.
- Une paire de clés SSH est facultative, mais elle peut vous aider à vous connecter à votre serveur maître Puppet si vous devez réinitialiser le mot de passe de l'administrateur de la console. Pour plus d'informations sur la création d'une paire de clés SSH, consultez Amazon EC2 Key Pairs dans le guide de l' EC2 utilisateur Amazon.
- Pour utiliser un domaine personnalisé, ajoutez les paramètres suivants à votre commande. Sinon, le processus de création d'un maître Puppet génère automatiquement un point de terminaison pour vous. Les trois paramètres sont requis pour configurer un domaine personnalisé. Pour plus d'informations sur les exigences supplémentaires relatives à l'utilisation de ces paramètres, consultez CreateServerla référence de l'API AWS OpsWorks CM.
  - `--custom-domain` - Point de terminaison public facultatif d'un serveur, tel que `https://aws.my-company.com`.
  - `--custom-certificate` - Un certificat HTTPS au format PEM. La valeur peut être un seul certificat auto-signé ou une chaîne de certificats.
  - `--custom-private-key` - Une clé privée au format PEM pour se connecter au serveur à l'aide de HTTPS. La clé privée ne doit pas être chiffrée ; elle ne peut pas être protégée par un mot de passe ou une phrase passe.
- Une maintenance hebdomadaire du système est obligatoire. Vous devez spécifier des valeurs valides au format suivant : `DDD:HH:MM`. L'heure doit être exprimée en heure UTC (temps universel coordonné). Si vous ne spécifiez aucune valeur pour `--preferred-`

`maintenance-window`, la valeur par défaut est un intervalle aléatoire d'une heure le mardi, le mercredi ou le vendredi.

- Les valeurs valides pour `--preferred-backup-window` doivent être spécifiées dans l'un des formats suivants : `HH:MM` pour les sauvegardes quotidiennes ou `DDD:HH:MM` pour les sauvegardes hebdomadaires. L'heure indiquée est en heure UTC. La valeur par défaut est une heure de début quotidienne aléatoire. Pour annuler les sauvegardes quotidiennes, ajoutez le paramètre `--disable-automated-backup`.
- Pour`--security-group-ids`, entrez un ou plusieurs groupes de sécurité IDs, séparés par un espace.
- Pour `--subnet-ids`, entrez un ID de sous-réseau.

```
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
```

L'exemple suivant crée un maître Puppet qui utilise un domaine personnalisé.

```
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 pour Puppet Enterprise, il faut environ 15 minutes pour créer un nouveau serveur. Dans la sortie de la commande `create-server`, copiez la valeur de l'attribut `Endpoint`. Voici un exemple.

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

   Ne supprimez pas la sortie de la commande `create-server` ou ne fermez pas votre session shell, car la sortie peut contenir des informations importantes qui ne seront plus réaffichées. Pour obtenir les mots de passe et le kit de démarrage à partir des résultats de la commande `create-server`, passez à l'étape suivante.

5. Si vous avez choisi que OpsWorks Puppet Enterprise génère un mot de passe pour vous, vous pouvez l'extraire dans un format utilisable à partir des `create-server` résultats en utilisant un processeur JSON tel que jq. Une fois que vous avez installé jq, vous pouvez exécuter les commandes suivantes pour extraire le mot de passe administrateur et le kit de démarrage Puppet. Si vous n'avez pas fourni votre propre mot de passe à l'étape 3, veillez à sauvegarder le mot de passe administrateur extrait dans un emplacement sécurisé, mais pratique.

```
#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**
>
> Vous ne pouvez pas régénérer un nouveau kit de démarrage de serveur maître Puppet Starter Kit dans l' AWS Management Console. Lorsque vous créez un Puppet master à l'aide de AWS CLI, exécutez la `jq` commande précédente pour enregistrer le kit de démarrage codé en base64 dans les `create-server` résultats sous forme de fichier ZIP.

6. Si vous n'avez pas extrait le kit de démarrage à partir des résultats des `create-server` commandes, vous pouvez éventuellement en télécharger un nouveau depuis la page Propriétés du serveur dans la console OpsWorks pour Puppet Enterprise.
7. Si vous n'utilisez pas un domaine personnalisé, passez à l'étape suivante. Si vous utilisez un domaine personnalisé avec le serveur, créez une entrée CNAME dans l'outil de gestion DNS de votre entreprise pour faire pointer votre domaine personnalisé vers le point de terminaison OpsWorks pour Puppet Enterprise que vous avez copié à l'étape 4. Vous ne pouvez pas accéder ou vous connecter à un serveur avec un domaine personnalisé tant que vous n'avez pas effectué cette étape.
8. Lorsque le processus de création du serveur est terminé, accédez à Configurer le serveur maître Puppet à l'aide du Starter Kit.

consultez aussi

- Créez un Puppet Enterprise Master à l'aide du AWS CLI
- Sauvegarder et restaurer un serveur OpsWorks pour Puppet Enterprise
- CreateServerdans le manuel de référence de l'API AWS OpsWorks CM
- create-server dans la référence de commande de l'AWS CLI

Utilisation des balises sur les AWS OpsWorks for Puppet Enterprise ressources

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les balises sont des mots ou des expressions qui jouent le rôle de métadonnées pour identifier et organiser vos ressources AWS. Dans OpsWorks Puppet Enterprise, une ressource peut comporter jusqu'à 50 balises appliquées par l'utilisateur. Chaque balise est constituée d'une clé et d'une valeur

facultative. Vous pouvez appliquer des balises aux ressources suivantes dans OpsWorks for Puppet Enterprise :

- OpsWorks pour les serveurs Puppet Enterprise
- Sauvegardes de OpsWorks pour serveurs Puppet Enterprise

Les balises sur les AWS ressources peuvent vous aider à suivre les coûts, à contrôler l'accès aux ressources, à regrouper les ressources pour automatiser les tâches ou à organiser les ressources par objectif ou par étape du cycle de vie. Pour de plus amples informations sur les avantages des balises, veuillez consulter la page relative aux stratégies de balisage AWS dans AWS Answers et Utilisation des balises de répartition des coûts dans le Guide de l'utilisateur AWS Billing and Cost Management .

Pour utiliser des balises afin de contrôler l'accès OpsWorks aux serveurs ou aux sauvegardes de Puppet Enterprise, créez ou modifiez des déclarations de politique dans AWS Identity and Access Management (IAM). Pour de plus amples informations, veuillez consulter Contrôle de l'accès aux ressources AWS à l'aide des balises de ressources dans le Guide de l'utilisateur AWS Identity and Access Management .

Lorsque vous appliquez des balises à un master OpsWorks for Puppet Enterprise, elles sont également appliquées aux sauvegardes du maître, au compartiment Amazon S3 qui stocke les sauvegardes, à l' EC2 instance Amazon du maître, aux secrets du maître qui y sont stockés et à AWS Secrets Manager l'adresse IP élastique utilisée par le maître. Les balises ne sont pas propagées vers la AWS CloudFormation pile AWS OpsWorks utilisée pour créer votre Puppet master.

Rubriques

- Comment fonctionnent les tags dans AWS OpsWorks for Puppet Enterprise
- Ajouter et gérer des tags dans OpsWorks Puppet Enterprise (console)
- Ajouter et gérer des balises dans OpsWorks Puppet Enterprise (CLI)
- consultez aussi

Comment fonctionnent les tags dans AWS OpsWorks for Puppet Enterprise

Dans cette version, vous pouvez ajouter et gérer des balises à l'aide de l'API AWS OpsWorks CM ou de l' AWS Management Console. AWS OpsWorks CM tente également d'ajouter des balises que vous ajoutez à un serveur aux AWS ressources associées au serveur, notamment l' EC2 instance, les secrets dans Secrets Manager, l'adresse IP Elastic, le groupe de sécurité, le compartiment S3 et les sauvegardes.

Le tableau suivant fournit une vue d'ensemble de la façon dont vous ajoutez et gérez les balises dans OpsWorks Puppet Enterprise.

Action	Ce qu'il faut utiliser
Ajoutez des balises à un nouveau serveur OpsWorks pour Puppet Enterprise ou à une sauvegarde que vous créez manuellement.	• Choisissez Créer un serveur Puppet Enterprise et ajoutez des balises dans la page Configurer les paramètres avancés. • Choisissez Créer une sauvegarde sur la page Sauvegardes d'un serveur existant et ajoutez des balises sur la page Créer une sauvegarde de votre serveur Puppet Enterprise. • Ajoutez un paramètre `Tags` aux commandes `CreateServer` ou `CreateBackup`.
Affichez les balises sur une ressource.	• Sur la page de détails de votre serveur, choisissez Balises dans le panneau de navigation. • Sur la page Sauvegardes de votre serveur, sélectionnez une sauvegarde, puis choisissez Modifier la sauvegarde. • Exécutez la commande `ListTagsForResource`.
Ajoutez des balises à un serveur existant OpsWorks pour Puppet Enterprise ou à une sauvegarde, que la sauvegarde ait été créée manuellement ou automatiquement.	• Sur la page de détails de votre serveur, choisissez Balises dans le panneau de navigation, puis choisissez Modifier. • Sur la page Sauvegardes de votre serveur, sélectionnez une sauvegarde, puis choisissez Modifier la sauvegarde. • Exécutez la commande `TagResource`.
Supprimez des balises d'une ressource.	• Sur la page de détails de votre serveur, choisissez Balises dans le panneau de navigation, puis choisissez Modifier.

Action	Ce qu'il faut utiliser
	Choisissez X en regard des balises que vous souhaitez supprimer. • Sur la page Sauvegardes de votre serveur, sélectionnez une sauvegarde, puis choisissez Modifier la sauvegarde. Choisissez X en regard des balises que vous souhaitez supprimer. • Exécutez la commande `UntagResource`.

Les réponses `DescribeServers` et `DescribeBackups` n'incluent pas d'informations sur les balises. Pour afficher les balises, utilisez l'API `ListTagsForResource`.

Ajouter et gérer des tags dans OpsWorks Puppet Enterprise (console)

Les procédures de cette section sont effectuées dans l' AWS Management Console.

Si vous ajoutez des balises, une clé de balise ne peut pas être vide. La clé peut comporter 127 caractères maximum et ne peut pas contenir uniquement des lettres Unicode, des chiffres ou des séparateurs, ni les caractères spéciaux suivants : `+ - = . _ : / @`. Une valeur de balise est facultative. Vous pouvez ajouter une balise ayant une clé, mais aucune valeur. La valeur peut comporter 255 caractères maximum et ne peut contenir que des lettres Unicode, des chiffres, des séparateurs ou les caractères spéciaux suivants : `+ - = . _ : / @`.

Rubriques

- Ajouter des balises à un nouveau serveur OpsWorks pour Puppet Enterprise (console)
- Ajouter des balises à une nouvelle sauvegarde (Console)
- Ajouter ou afficher des balises sur un serveur existant (Console)
- Ajouter ou afficher des balises sur une sauvegarde existante (Console)
- Supprimer les balises d'un serveur (Console)
- Supprimer les balises d'une sauvegarde (Console)

Ajouter des balises à un nouveau serveur OpsWorks pour Puppet Enterprise (console)

1. Assurez-vous de remplir toutes les conditions requises pour créer un master OpsWorks pour Puppet Enterprise.
2. Suivez les étapes 1 à 8 dans Créez un Puppet Enterprise Master à l'aide du AWS Management Console.
3. Après avoir spécifié les paramètres de sauvegarde automatisée, ajoutez des balises dans la zone Balises de la page Configurer les paramètres avancés. Vous pouvez ajouter un maximum de 50 balises. Lorsque vous avez fini d'ajouter des balises, choisissez Suivant.
4. Passez à l'étape 11 de Créez un Puppet Enterprise Master à l'aide du AWS Management Console, et passez en revue les paramètres que vous avez choisis pour le nouveau serveur.

Ajouter des balises à une nouvelle sauvegarde (Console)

1. Sur la page OpsWorks d'accueil de Puppet Enterprise, choisissez un Puppet master existant.
2. Dans la page de détails du serveur, choisissez Sauvegardes dans le panneau de navigation.
3. Sur la page Sauvegardes choisissez Créer une sauvegarde.
4. Ajouter des balises. Une fois que vous avez fini d'ajouter des règles, choisissez Créer.

Ajouter ou afficher des balises sur un serveur existant (Console)

1. Sur la OpsWorks page d'accueil de Puppet Enterprise, choisissez un Puppet master existant pour ouvrir sa page de détails.
2. Choisissez Balises dans le panneau de navigation ou, en bas de la page de détails, choisissez Afficher toutes les balises.
3. Sur la page Balises, choisissez Modifier.
4. Ajoutez ou modifiez des balises sur le serveur. Lorsque vous avez terminé, choisissez Save (Enregistrer)

> **Note**
>
> Sachez que la modification des balises sur votre Puppet Master modifie également les balises des ressources associées au serveur, telles que l' EC2 instance, l'adresse IP élastique, le groupe de sécurité, le compartiment S3 et les sauvegardes.

Ajouter ou afficher des balises sur une sauvegarde existante (Console)

1. Sur la OpsWorks page d'accueil de Puppet Enterprise, choisissez un Puppet master existant pour ouvrir sa page de détails.
2. Choisissez Sauvegardes dans le panneau de navigation ou, dans la zone Sauvegardes récentes de la page de détails, choisissez Afficher toutes les sauvegardes.
3. Sur la page Sauvegardes, choisissez une sauvegarde à gérer, puis choisissez Modifier la sauvegarde.
4. Ajoutez ou modifiez des balises sur la sauvegarde. Choisissez Mettre à jour une fois que vous avez terminé.

Supprimer les balises d'un serveur (Console)

1. Sur la OpsWorks page d'accueil de Puppet Enterprise, choisissez un Puppet master existant pour ouvrir sa page de détails.
2. Choisissez Balises dans le panneau de navigation ou, en bas de la page de détails, choisissez Afficher toutes les balises.
3. Sur la page Balises, choisissez Modifier.
4. Choisissez X en regard d'une balise pour la supprimer. Lorsque vous avez terminé, choisissez Save (Enregistrer)

> **Note**
>
> Sachez que la modification des balises sur votre Puppet Master modifie également les balises des ressources associées au serveur, telles que l' EC2 instance, l'adresse IP élastique, le groupe de sécurité, le compartiment S3 et les sauvegardes.

Supprimer les balises d'une sauvegarde (Console)

1. Sur la OpsWorks page d'accueil de Puppet Enterprise, choisissez un Puppet master existant pour ouvrir sa page de détails.
2. Choisissez Sauvegardes dans le panneau de navigation ou, dans la zone Sauvegardes récentes de la page de détails, choisissez Afficher toutes les sauvegardes.

3. Sur la page Sauvegardes, choisissez une sauvegarde à gérer, puis choisissez Modifier la sauvegarde.
4. Choisissez X en regard d'une balise pour la supprimer. Choisissez Mettre à jour une fois que vous avez terminé.

Ajouter et gérer des balises dans OpsWorks Puppet Enterprise (CLI)

Les procédures de cette section sont effectuées dans l' AWS CLI. Assurez-vous que vous utilisez la dernière version du AWS CLI avant de commencer à travailler avec les balises. Pour plus d'informations sur l'installation ou la mise à jour du AWS CLI, voir Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Si vous ajoutez des balises, une clé de balise ne peut pas être vide. La clé peut comporter 127 caractères maximum et ne peut pas contenir uniquement des lettres Unicode, des chiffres ou des séparateurs, ni les caractères spéciaux suivants : `+ - = . _ : / @`. Une valeur de balise est facultative. Vous pouvez ajouter une balise ayant une clé, mais aucune valeur. La valeur peut comporter 255 caractères maximum et ne peut contenir que des lettres Unicode, des chiffres, des séparateurs ou les caractères spéciaux suivants : `+ - = . _ : / @`.

Rubriques

- Ajouter des balises à un nouveau serveur OpsWorks pour Puppet Enterprise (CLI)
- Ajouter des balises à une nouvelle sauvegarde (interface de ligne de commande)
- Ajouter des balises à des serveurs ou sauvegardes existants (Interface de ligne de commande)
- Liste des balises de ressource (Interface de ligne de commande)
- Supprimez des balises d'une ressource (Interface de ligne de commande)

Ajouter des balises à un nouveau serveur OpsWorks pour Puppet Enterprise (CLI)

Vous pouvez utiliser le AWS CLI pour ajouter des balises lorsque vous créez un serveur OpsWorks pour Puppet Enterprise. Cette procédure ne décrit pas en détail la création d'un serveur. Pour des informations détaillées sur la création d'un serveur OpsWorks pour Puppet Enterprise à l'aide du AWS CLI, consultez Créez un Puppet Enterprise Master à l'aide du AWS CLI ce guide. Vous pouvez ajouter jusqu'à 50 balises à un serveur.

1. Assurez-vous de remplir toutes les conditions préalables à la création d'un serveur OpsWorks pour Puppet Enterprise.

2. Effectuez les étapes 1 à 4 de Créez un Puppet Enterprise Master à l'aide du AWS CLI.
3. Pour l'étape 5, lorsque vous exécutez la commande `create-server`, ajoutez le paramètre `--tags` à la commande, comme indiqué dans l'exemple suivant.

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

   Voici un exemple montrant uniquement la partie correspondant aux balises de la commande `create-server`.

   ```
   aws opsworks-cm create-server ... --tags Key=Stage,Value=Production
    Key=Department,Value=Marketing
   ```
4. Suivez les étapes restantes fournies dans Créez un Puppet Enterprise Master à l'aide du AWS CLI. Pour vérifier que vos balises ont été ajoutées au nouveau serveur, suivez les étapes décrites dans Liste des balises de ressource (Interface de ligne de commande) de la présente rubrique.

Ajouter des balises à une nouvelle sauvegarde (interface de ligne de commande)

Vous pouvez utiliser le AWS CLI pour ajouter des balises lorsque vous créez une nouvelle sauvegarde manuelle d'un serveur OpsWorks pour Puppet Enterprise. Cette procédure ne décrit pas en détail la création d'une sauvegarde manuelle. Pour obtenir des informations détaillées sur la création d'une sauvegarde manuelle, reportez-vous à « Pour effectuer une sauvegarde manuelle dans le AWS CLI » dansSauvegardez et OpsWorks pour Puppet Enterprise Server. Vous pouvez ajouter jusqu'à 50 balises à une sauvegarde. Si un serveur possède des balises, les nouvelles sauvegardes sont automatiquement balisées avec les balises du serveur.

Par défaut, lorsque vous créez un nouveau serveur OpsWorks pour Puppet Enterprise, les sauvegardes automatisées sont activées. Vous pouvez ajouter des balises à une sauvegarde automatisée en exécutant la commande `tag-resource` décrite dans Ajouter des balises à des serveurs ou sauvegardes existants (Interface de ligne de commande) de la présente rubrique.

- Pour ajouter des balises à une sauvegarde manuelle lors de la création de la sauvegarde, exécutez la commande suivante. Seule la partie de la commande correspondant aux balises est affichée. Pour obtenir un exemple de la commande `create-backup` complète, veuillez consulter Pour effectuer une sauvegarde manuelle dans l' AWS CLI" dans Sauvegardez et OpsWorks pour Puppet Enterprise Server.

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

L'exemple suivant montre uniquement la partie de la commande `create-backup` correspondant aux balises.

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

Ajouter des balises à des serveurs ou sauvegardes existants (Interface de ligne de commande)

Vous pouvez exécuter la `tag-resource` commande pour ajouter des balises aux serveurs ou aux sauvegardes existants OpsWorks pour Puppet Enterprise (que les sauvegardes aient été créées automatiquement ou manuellement). Spécifiez l'ARN (Amazon Resource Name) d'une ressource cible pour y ajouter des balises.

1. Pour obtenir l'ARN de la ressource à laquelle vous souhaitez appliquer des balises :

   - Pour un serveur, exécutez `describe-servers --server-name` *`server_name`*. Les résultats de la commande indiquent l'ARN du serveur.
   - Pour une sauvegarde, exécutez `describe-backups --backup-id` *`backup_ID`*. Les résultats de la commande montrent l'ARN de la sauvegarde. Vous pouvez également exécuter `describe-backups --server-name` *`server_name`* pour afficher des informations sur toutes les sauvegardes d'un serveur spécifique OpsWorks pour Puppet Enterprise.

   L'exemple suivant montre uniquement la valeur `ServerArn` dans les résultats d'une commande `describe-servers --server-name opsworks-cm-test`. La valeur `ServerArn` est ajoutée à une commande `tag-resource` pour ajouter des balises au serveur.

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

```
    ]
}
```

2. Exécutez la commande `tag-resource` avec l'ARN que vous avez renvoyé à l'étape 1.

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

Voici un exemple.

```
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. Pour vérifier que les balises ont été ajoutées avec succès, passez à la procédure suivante, Liste des balises de ressource (Interface de ligne de commande).

Liste des balises de ressource (Interface de ligne de commande)

Vous pouvez exécuter la `list-tags-for-resource` commande pour afficher les balises associées aux serveurs ou OpsWorks aux sauvegardes de Puppet Enterprise. Spécifiez l'ARN d'une ressource cible pour afficher ses balises.

1. Pour obtenir l'ARN de la ressource pour laquelle vous souhaitez répertorier les balises :

   - Pour un serveur, exécutez `describe-servers --server-name` *`server_name`*. Les résultats de la commande indiquent l'ARN du serveur.
   - Pour une sauvegarde, exécutez `describe-backups --backup-id` *`backup_ID`*. Les résultats de la commande montrent l'ARN de la sauvegarde. Vous pouvez également exécuter `describe-backups --server-name` *`server_name`* pour afficher des informations sur toutes les sauvegardes d'un serveur spécifique OpsWorks pour Puppet Enterprise.

2. Exécutez la commande `list-tags-for-resource` avec l'ARN que vous avez renvoyé à l'étape 1.

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

Voici un exemple.

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

Si la ressource comporte des balises, la commande renvoie des résultats similaires au résultat suivant.

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

Supprimez des balises d'une ressource (Interface de ligne de commande)

Vous pouvez exécuter la `untag-resource` commande pour supprimer les balises des serveurs ou OpsWorks des sauvegardes de Puppet Enterprise. Si la ressource est supprimée, les balises de la ressource sont également supprimées. Spécifiez l'ARN (Amazon Resource Name) d'une ressource cible pour en supprimer les balises.

1. Pour obtenir l'ARN de la ressource pour laquelle vous souhaitez supprimer des balises :
   - Pour un serveur, exécutez `describe-servers --server-name` *`server_name`*. Les résultats de la commande indiquent l'ARN du serveur.
   - Pour une sauvegarde, exécutez `describe-backups --backup-id` *`backup_ID`*. Les résultats de la commande montrent l'ARN de la sauvegarde. Vous pouvez également exécuter `describe-backups --server-name` *`server_name`* pour afficher des informations sur toutes les sauvegardes d'un serveur spécifique OpsWorks pour Puppet Enterprise.
2. Exécutez la commande `untag-resource` avec l'ARN que vous avez renvoyé à l'étape 1. Spécifiez uniquement les balises que vous souhaitez supprimer.

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

Dans cet exemple, la commande `untag-resource` supprime uniquement la balise dont la clé est `Stage` et dont la valeur est `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. Pour vérifier que les balises ont été supprimées avec succès, suivez les étapes décrites dans Liste des balises de ressource (Interface de ligne de commande) de la présente rubrique.

consultez aussi

- Créez un Puppet Enterprise Master à l'aide du AWS CLI
- Sauvegardez et OpsWorks pour Puppet Enterprise Server
- Stratégies de balisage AWS
- Contrôle de l'accès aux AWS ressources à l'aide des balises de ressources dans le guide de AWS Identity and Access Management l'utilisateur
- Utilisation des balises de répartition des coûts dans le Guide de l'utilisateur AWS Billing and Cost Management .
- CreateBackup dans la Référence d'API AWS OpsWorks CM
- CreateServer dans la Référence d'API AWS OpsWorks CM
- TagResource dans la Référence d'API AWS OpsWorks CM
- ListTagsForResource dans la Référence d'API AWS OpsWorks CM
- UntagResource dans la Référence d'API AWS OpsWorks CM

Sauvegarder et restaurer un serveur OpsWorks pour Puppet Enterprise

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section décrit la sauvegarde et la restauration d'un serveur OpsWorks for Puppet Enterprise.

Rubriques

- Sauvegardez et OpsWorks pour Puppet Enterprise Server
- Restaurer un serveur OpsWorks for Puppet Enterprise à partir d'une sauvegarde

Sauvegardez et OpsWorks pour Puppet Enterprise Server

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez définir une fréquence quotidienne ou hebdomadaire OpsWorks pour la sauvegarde du serveur Puppet Enterprise et demander au service de stocker les sauvegardes dans Amazon Simple Storage Service (Amazon S3) en votre nom. Sinon, vous pouvez faire effectuer des sauvegardes manuelles à la demande.

Les sauvegardes étant stockées dans Amazon S3, elles entraînent des frais supplémentaires. Vous pouvez définir une période de rétention des sauvegardes de 30 générations maximum. Vous

pouvez soumettre une demande de service pour faire modifier cette limite en utilisant les canaux d' AWS assistance. Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Vous pouvez ajouter des balises aux sauvegardes d'un master OpsWorks pour Puppet Enterprise. Si vous avez ajouté des balises à un maître OpsWorks pour Puppet Enterprise, les sauvegardes automatiques du maître Puppet héritent de ces balises. Pour de plus amples informations sur l'ajout et la gestion de balises sur les sauvegardes, veuillez consulter Utilisation des balises sur les AWS OpsWorks for Puppet Enterprise ressources dans ce guide.

Rubriques

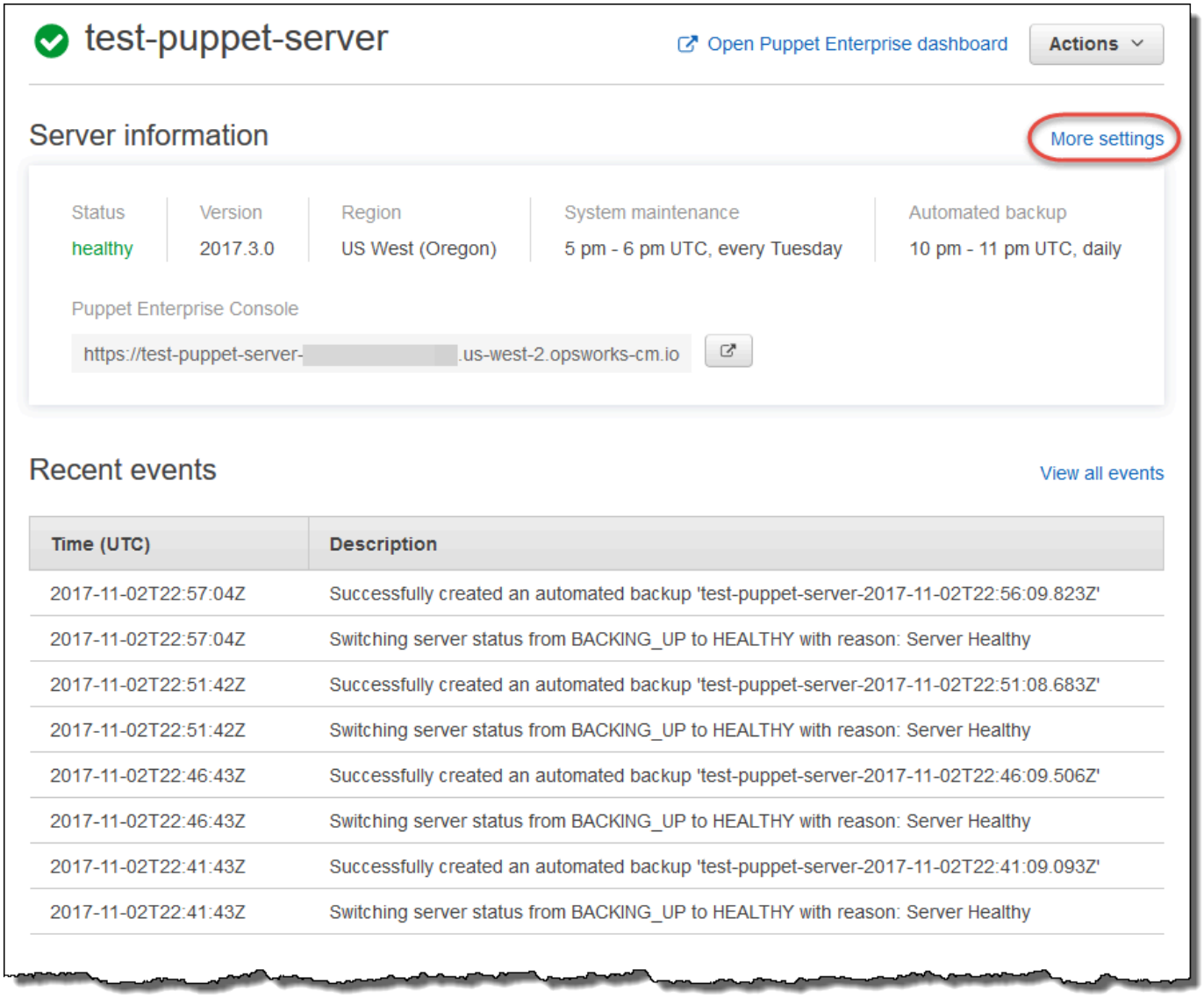
- Sauvegardes automatisées
- Sauvegardes manuelles
- Supprimer des sauvegardes

Sauvegardes automatisées

Lorsque vous configurez votre serveur OpsWorks pour Puppet Enterprise, vous choisissez des sauvegardes automatisées ou manuelles. OpsWorks for Puppet Enterprise lance des sauvegardes automatisées pendant l'heure et le jour que vous choisissez dans la section Sauvegarde automatisée de la page Configurer les paramètres avancés de l'assistant de configuration. Une fois votre serveur en ligne, vous pouvez modifier ses paramètres de sauvegarde en effectuant les étapes suivantes sur sa page de propriétés.

Pour modifier les paramètres de sauvegarde automatisée

1. Sur la page de propriétés du serveur, choisissez More settings (Plus de paramètres).



2. Pour désactiver les sauvegardes automatisées, choisissez Non pour l'option Activer les sauvegardes automatiques. Enregistrez'vos modifications ; vous n'avez pas besoin de passer à l'étape suivante.
3. Dans la section Sauvegarde automatique, modifiez la fréquence, l'heure de début ou les générations à conserver. Enregistrez vos modifications.

Sauvegardes manuelles

Vous pouvez démarrer une sauvegarde manuelle à tout moment dans le ou AWS Management Console en exécutant la commande AWS CLI create-backup. Les sauvegardes manuelles ne sont pas incluses dans les 30 générations maximum de sauvegardes automatisées stockées. Un

maximum de 10 sauvegardes manuelles sont stockées et doivent être supprimées manuellement d'Amazon S3.

Pour effectuer une sauvegarde manuelle dans le AWS Management Console

1. Sur la page Puppet Enterprise servers (serveurs Puppet Enterprise), sélectionnez le serveur à sauvegarder.
2. Sur la page de propriétés du serveur, sélectionnez Sauvegardes dans le panneau de navigation de gauche.
3. Choisissez Créer une sauvegarde.
4. La sauvegarde manuelle est terminée lorsque la page affiche une coche verte dans la colonne Statut de la sauvegarde.

Pour effectuer une sauvegarde manuelle dans le AWS CLI

Vous pouvez ajouter des balises lorsque vous créez une sauvegarde manuelle d'un OpsWorks serveur Puppet Enterprise. Pour de plus amples informations sur l'ajout de balises lors de la création d'une sauvegarde manuelle, reportez-vous à Ajouter des balises à une nouvelle sauvegarde (interface de ligne de commande).

- Pour démarrer une sauvegarde manuelle, exécutez la AWS CLI commande suivante.

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

Supprimer des sauvegardes

Si vous supprimez une sauvegarde, celle-ci est définitivement éliminée du compartiment S3 où les sauvegardes sont stockées.

Pour supprimer une sauvegarde dans AWS Management Console

1. Sur la page Puppet Enterprise servers (serveurs Puppet Enterprise), sélectionnez le serveur à sauvegarder.
2. Sur la page de propriétés du serveur, sélectionnez Sauvegardes dans le panneau de navigation de gauche.

3. Sélectionnez la sauvegarde que vous voulez supprimer et choisissez Supprimer une sauvegarde. Vous ne pouvez sélectionner qu'une seule sauvegarde à la fois.
4. Lorsque vous êtes invité à confirmer la suppression, cochez la case Delete the backup, which is stored in an S3 bucket (Supprimer la sauvegarde stockée dans un compartiment S3), puis sélectionnez Oui, supprimer.

Pour supprimer une sauvegarde dans AWS CLI

- Pour supprimer une sauvegarde, exécutez la AWS CLI commande suivante en remplaçant la valeur de `--backup-id` par l'ID de la sauvegarde que vous souhaitez supprimer. IDs Les sauvegardes sont au format*`ServerName-yyyyMMddHHmmssSSS`*. Par exemple, **`puppet-server-20171218132604388`**.

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

Restaurer un serveur OpsWorks for Puppet Enterprise à partir d'une sauvegarde

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Après avoir parcouru les sauvegardes disponibles, vous pouvez facilement choisir un moment à partir duquel vous souhaitez restaurer votre serveur OpsWorks pour Puppet Enterprise. Les sauvegardes de serveur contiennent des données logicielles persistantes de gestion de la configuration, notamment des informations sur les modules, associations et bases de données (y compris rapports, faits, etc.). La restauration sur place d'un serveur (c'est-à-dire la restauration du serveur existant OpsWorks pour Puppet Enterprise sur une nouvelle EC2 instance) permet de réenregistrer les nœuds enregistrés au moment de la sauvegarde que vous utilisez pour restaurer le serveur, et de transférer le trafic vers la nouvelle instance si la restauration est réussie et si l'état du serveur

restauré OpsWorks pour Puppet Enterprise est le cas. `Healthy` La restauration vers un serveur nouvellement créé OpsWorks pour Puppet Enterprise ne permet pas de maintenir les connexions aux nœuds. La restauration d'un serveur ne permet pas de mettre à jour la version du logiciel Puppet. En effet, elle applique les versions de Puppet et les données de gestion de la configuration qui sont disponibles dans la sauvegarde que vous choisissez.

La restauration d'un serveur prend généralement plus de temps que la création d'un nouveau serveur ; le temps dépend de la taille de la sauvegarde que vous choisissez. Une fois la restauration terminée, l'ancienne EC2 instance reste à `Stopped` l'état `Running` or, mais uniquement temporairement. À terme, elle est résiliée.

Dans cette version, vous pouvez utiliser le AWS CLI pour restaurer un Puppet Master dans OpsWorks Puppet Enterprise.

> **Note**
>
> Vous pouvez également exécuter la commande restore-server pour modifier le type d'instance actuelle, ou pour restaurer ou définir votre clé SSH si elles est perdue ou divulguée.

Pour restaurer un serveur à partir d'une sauvegarde

1. Dans le AWS CLI, exécutez la commande suivante pour renvoyer une liste des sauvegardes disponibles et leurs IDs. Notez l'ID de la sauvegarde que vous souhaitez utiliser. IDs Les sauvegardes sont au format*myServerName-yyyyMMddHHmmssSSS*.

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

2. Exécutez la commande suivante.

   ```
   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"
   ```

   Voici un exemple.

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

3. Attendez que la restauration soit terminée.

Maintenance du système dans OpsWorks Puppet Enterprise

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

La maintenance obligatoire du système garantit que les dernières versions AWS testées de Puppet Server, y compris les mises à jour de sécurité, fonctionnent toujours sur et OpsWorks pour le serveur Puppet Enterprise. La maintenance du système doit avoir lieu au moins une fois par semaine. En utilisant le AWS CLI, vous pouvez configurer la maintenance automatique quotidienne, si vous le souhaitez. Vous pouvez également utiliser le AWS CLI pour effectuer la maintenance du système à la demande, en plus de la maintenance planifiée du système.

Lorsque de nouvelles versions du logiciel Puppet deviennent disponibles, la maintenance système est conçue pour mettre à jour automatiquement la version de Puppet Server sur le serveur dès que celle-ci réussit les tests AWS. AWS effectue des tests approfondis pour vérifier que les mises à niveau de Puppet sont prêtes pour la production et ne perturbent pas les environnements clients existants. Il peut donc y avoir des décalages entre les versions du logiciel Puppet et leur disponibilité en tant qu'applications sur les serveurs Puppet Enterprise existants OpsWorks . Pour mettre à jour les versions du logiciel Puppet à la demande, consultez Démarrage de la maintenance du système à la demande dans cette rubrique.

La maintenance du système lance une nouvelle instance à partir d'une sauvegarde effectuée dans le cadre du processus de maintenance, ce qui permet de réduire les risques liés à la dégradation ou à l'altération des EC2 instances Amazon soumises à une maintenance périodique.

> **Important**
>
> La maintenance du système supprime tous les fichiers ou configurations personnalisées que vous avez ajoutés au serveur OpsWorks for Puppet Enterprise. Pour plus d'informations sur

la résolution d'un problème de perte de configuration ou de fichier, consultez Restauration des configurations personnalisées et des fichiers après la maintenance dans cette rubrique.

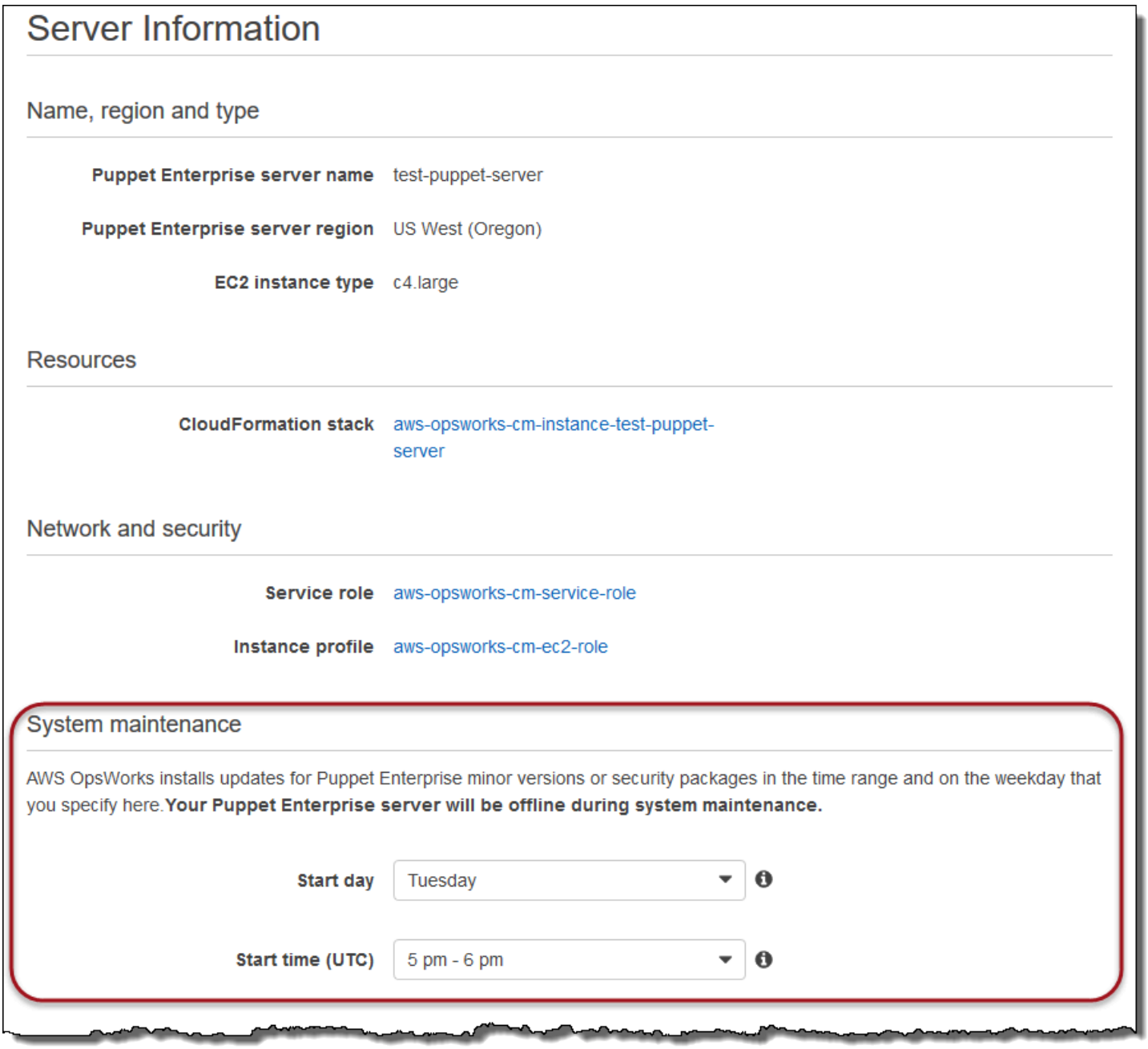
Rubriques

- Configuration de la maintenance du système
- Démarrage de la maintenance du système à la demande
- Restauration des configurations personnalisées et des fichiers après la maintenance

Configuration de la maintenance du système

Lorsque vous créez un nouveau serveur OpsWorks pour Puppet Enterprise, vous pouvez configurer un jour et une heure de la semaine, en temps universel coordonné (UTC), pour le début de la maintenance du système. La maintenance commence au cours de l'heure que vous spécifiez. Etant donné que le serveur doit être hors ligne lors de la maintenance du système, choisissez une heure où le serveur est peu sollicité pendant les heures normales de bureau. L'état du serveur est `UNDER_MAINTENANCE` lorsque la maintenance est en cours.

Vous pouvez également modifier les paramètres de maintenance du système sur un serveur existant OpsWorks pour Puppet Enterprise, en modifiant les paramètres dans la zone Maintenance du système de la page Paramètres de votre serveur, comme indiqué dans la capture d'écran suivante.



Dans la section System maintenance (Maintenance du système), définissez le jour et l'heure de début de la maintenance du système.

Configuration de la maintenance du système à l'aide du AWS CLI

Vous pouvez également configurer le démarrage automatique du système de maintenance à l'aide de l' AWS CLI. Vous AWS CLI permet de configurer la maintenance automatique quotidienne, si vous le souhaitez, en omettant le préfixe à trois caractères du jour de la semaine.

Dans une commande `create-server`, ajoutez le paramètre `--preferred-maintenance-window` à votre commande, après avoir indiqué les exigences relatives à la création de l'instance serveur (par exemple, le type d'instance, l'ARN du profil d'instance et l'ARN du rôle de service). Dans l'exemple `create-server` suivant, la valeur définie pour `--preferred-maintenance-window` est `Mon:08:00`, ce qui signifie que vous avez défini le début de la maintenance chaque lundi matin à 8 h 00. UTC.

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

Dans une commande `update-server`, vous pouvez mettre à jour la valeur `--preferred-maintenance-window` seulement, si vous le souhaitez. Dans l'exemple suivant, la fenêtre de maintenance est définie à vendredi soir à 18 h 15. UTC.

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

Pour modifier l'heure de début de la fenêtre de maintenance à 18 h 15 chaque jour, n'indiquez pas le préfixe à trois caractères du jour de la semaine, comme illustré dans l'exemple suivant.

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

Pour plus d'informations sur la définition de la fenêtre de maintenance système préférée à l'aide du AWS CLI, voir create-server et update-server.

Démarrage de la maintenance du système à la demande

Pour démarrer la maintenance du système à la demande, en dehors de votre maintenance automatique hebdomadaire ou quotidienne configurée, exécutez la AWS CLI commande suivante. Vous ne pouvez pas démarrer la maintenance à la demande dans AWS Management Console.

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

Pour plus d'informations sur cette commande, consultez start-maintenance.

Restauration des configurations personnalisées et des fichiers après la maintenance

La maintenance du système peut supprimer ou modifier les fichiers personnalisés ou les configurations que vous avez ajoutés à votre serveur OpsWorks pour Puppet Enterprise.

Si, après une opération de maintenance, il manque à votre Puppet Master des fichiers ou des paramètres que vous avez ajoutés à l'aide `RunCommand` de SSH, vous pouvez utiliser une Amazon Machine Image (AMI) pour lancer une nouvelle instance Amazon EC2 . AMIs sont disponibles et sont créés à partir de la configuration préalable à la maintenance d'un serveur.

L'état de la nouvelle instance correspond à celui du maître Puppet avant la maintenance. La nouvelle instance doit normalement inclure vos fichiers et paramètres manquants.

> **Important**
>
> Vous ne pouvez pas utiliser la nouvelle instance pour restaurer votre serveur ; l'instance ne peut pas être exécutée en tant que maître Puppet. Vous pouvez utiliser l'instance uniquement pour récupérer vos fichiers et les paramètres de configuration.

Pour lancer une EC2 instance depuis une AMI, dans la EC2 console Amazon, ouvrez l'assistant de lancement, choisissez My AMIs, puis choisissez l'AMI qui porte le nom de votre serveur. Suivez les étapes de EC2 l'assistant Amazon comme vous le feriez pour tout autre lancement d'instance.

Ajout automatique de nœuds dans OpsWorks Puppet Enterprise

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette rubrique explique comment ajouter automatiquement des nœuds Amazon Elastic Compute Cloud (Amazon EC2) à votre serveur OpsWorks pour Puppet Enterprise. Dans la section Ajouter des

nœuds à gérer par le serveur maître Puppet, vous avez appris à utiliser la commande `associate-node` pour ajouter un nœud à la fois à votre serveur Puppet Enterprise. Le code présenté dans cette rubrique montre comment ajouter automatiquement des nœuds à l'aide de la méthode sans surveillance. La méthode recommandée pour associer de nouveaux nœuds sans surveillance (ou automatique) consiste à configurer les données EC2 utilisateur Amazon. Par défaut, un serveur OpsWorks pour Puppet Enterprise est déjà `puppet-agent`disponible pour les systèmes d'exploitation des nœuds Ubuntu, Amazon Linux et RHEL.

Pour plus d'informations sur la façon de dissocier un nœud, consultez Dissocier un nœud d'un serveur OpsWorks for Puppet Enterprise ce guide et la documentation de `disassociate-node`l'API OpsWorks for Puppet Enterprise.

Étape 1 : créer un rôle IAM à utiliser comme profil d'instance

Créez un rôle AWS Identity and Access Management (IAM) à utiliser comme profil d' EC2 instance et associez la politique suivante au rôle IAM. Cette politique permet à l'`opsworks-cm`API de communiquer avec l' EC2 instance lors de l'enregistrement du nœud. Pour plus d'informations sur les profils d'instance, consultez la section Utilisation des profils d'instance dans la EC2 documentation Amazon. Pour plus d'informations sur la création d'un rôle IAM, consultez la section Création d'un rôle IAM dans la console dans la documentation Amazon EC2 .

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

Étape 2 : Création d'instances en utilisant un script d'association sans surveillance

Pour créer des EC2 instances, vous pouvez copier le script de données utilisateur inclus dans le kit de démarrage dans la `userdata` section des instructions relatives aux EC2 instances, dans les configurations de lancement du groupe Amazon EC2 Auto Scaling ou dans un AWS CloudFormation modèle. Le script n'est pris en charge que pour EC2 les instances exécutant les systèmes d'exploitation Ubuntu et Amazon Linux. Pour plus d'informations sur l'ajout de scripts aux données utilisateur, consultez Exécuter des commandes sur votre instance Linux au lancement dans la EC2 documentation Amazon. Le moyen le plus simple de créer un nouveau nœud consiste à utiliser l'assistant de lancement d' EC2instance Amazon. Cette procédure pas à pas utilise l'exemple de configuration du module du serveur Web Apache décrit dansCommencer à utiliser OpsWorks pour Puppet Enterprise.

1. Le script de données utilisateur dans le Starter Kit exécute la commande d'API `opsworks-cm associate-node` pour associer un nouveau nœud à votre maître Puppet. Dans cette version, il installe également la version actuelle de AWS CLI on the node pour vous, au cas où elle n'exécute pas déjà la up-to-date version la plus récente. Enregistrez ce script à un emplacement approprié (ex. : `userdata.sh`).

   Par défaut, le nom du nouveau nœud enregistré correspond à l'ID de l'instance.
2. Suivez la procédure décrite dans la section Lancement d'une instance dans la EC2 documentation, avec les modifications ici. Dans l'assistant de lancement d' EC2 instance, choisissez une AMI Amazon Linux.
3. Sur la page Configure Instance Details, sélectionnez myPuppetinstanceprofile, le rôle que vous avez créé à la section Étape 1 : créer un rôle IAM à utiliser comme profil d'instance, comme étant votre rôle IAM.
4. Dans la zone Advanced Details, chargez le script `userdata.sh` que vous avez créé à l'étape 1.
5. Aucune modification n'est nécessaire sur la page Add Storage. Accédez à la page Add Tags.

   En appliquant des balises à votre EC2 instance, vous pouvez personnaliser le comportement de`userdata.sh`. Dans cet exemple, appliquez le rôle `apache_webserver` à votre nœud en ajoutant la balise suivante : **`pp_role`** avec la valeur **`apache_webserver`**.

   En affectant la valeur `pp_role` au nœud, vous définissez des valeurs de données qui sont stockées de manière permanente dans le certificat de l'agent du nœud, ce qui permet de classer le nœud parmi les nœuds de confiance. Pour plus d'informations, consultez la section

Extension requests (permanent certificate data) [Extension des requêtes (données de certificat permanentes)] dans la documentation de la plate-forme Puppet.

6. Sur la page Configurer le groupe de sécurité, choisissez Ajouter une règle, puis choisissez le type HTTP pour ouvrir le port 8080 pour le serveur Web Apache dans cet exemple.
7. Sélectionnez Review and Launch, puis Launch. Lorsque votre nouveau nœud démarre, il applique la configuration Apache du module d'exemple dans lequel vous l'avez configuréExemple de configuration du kit de démarrage Apache.
8. Lorsque vous ouvrez la page Web liée au DNS public de votre nouveau nœud, vous devriez voir un site Web hébergé par votre serveur Web Apache géré par Puppet.

Dissocier un nœud d'un serveur OpsWorks for Puppet Enterprise

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section décrit comment dissocier ou supprimer un nœud géré de la gestion par un serveur OpsWorks pour Puppet Enterprise. Cette opération est effectuée sur la ligne de commande ou dans la console Puppet Enterprise ; vous ne pouvez pas dissocier les nœuds dans la console de gestion OpsWorks de Puppet Enterprise. Actuellement, l'API OpsWorks for Puppet Enterprise n'autorise pas la suppression par lots de plusieurs nœuds. La commande de cette section dissocie un nœud à la fois.

Nous vous recommandons de dissocier les nœuds du maître Puppet avant de supprimer le serveur. Ainsi, les nœuds peuvent continuer de fonctionner sans tenter de se reconnecter au serveur. Pour ce faire, exécutez la `disassociate-node` AWS CLI commande. Pour supprimer complètement un nœud à partir de PE, vous devez dissocier le nœud et révoquer son certificat. De cette manière, le nœud ne cherchera pas en permanence à se signaler auprès du maître Puppet. Vous devez également procéder à la désinstallation `puppet-agent` sur les nœuds lorsque vous ne souhaitez plus les gérer en utilisant le maître Puppet.

Pour dissocier des nœuds

1. Dans le AWS CLI, exécutez la commande suivante pour dissocier les nœuds. *Node_name*est le nom du nœud que vous souhaitez dissocier ; pour les EC2 instances Amazon, il s'agit de l'ID de l'instance. *Server_name*est le nom du Puppet Master dont vous souhaitez dissocier le nœud. Les deux paramètres sont requis. Le paramètre `--region` n'est pas obligatoire, sauf si vous souhaitez dissocier un nœud d'un maître Puppet qui n'est pas dans votre région par défaut.

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

   Voici un exemple de commande.

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

2. Attendez qu'un message de réponse indique que la dissociation est terminée.

Pour plus d'informations sur la façon de supprimer un serveur OpsWorks pour Puppet Enterprise, consultezSupprimer un OpsWorks pour Puppet Enterprise Server.

consultez aussi

- Suppression de nœuds dans la documentation de Puppet Enterprise.

Supprimer un OpsWorks pour Puppet Enterprise Server

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section explique comment supprimer un serveur OpsWorks pour Puppet Enterprise. Si vous supprimez un serveur, cela supprime également les événements, les journaux et les éventuels

modules qui y étaient stockés. Les ressources de support (instance Amazon Elastic Compute Cloud, volume Amazon Elastic Block Store, etc.) sont également supprimées, ainsi que toutes les sauvegardes automatisées.

Même si la suppression d'un serveur ne supprime pas de nœuds, ces derniers ne sont plus gérés par le serveur supprimé et tenteront continuellement de se reconnecter. C'est pourquoi nous recommandons de dissocier les nœuds gérés avant de supprimer un maître Puppet. Dans cette version, vous pouvez dissocier les nœuds en exécutant une AWS CLI commande.

Étape 1 : Dissociation de nœuds gérés

Dissociez les nœuds du maître Puppet avant de supprimer le serveur pour que les nœuds continuent de fonctionner sans tenter de se reconnecter au serveur. Pour ce faire, exécutez la `disassociate-node` AWS CLI commande.

Pour dissocier des nœuds

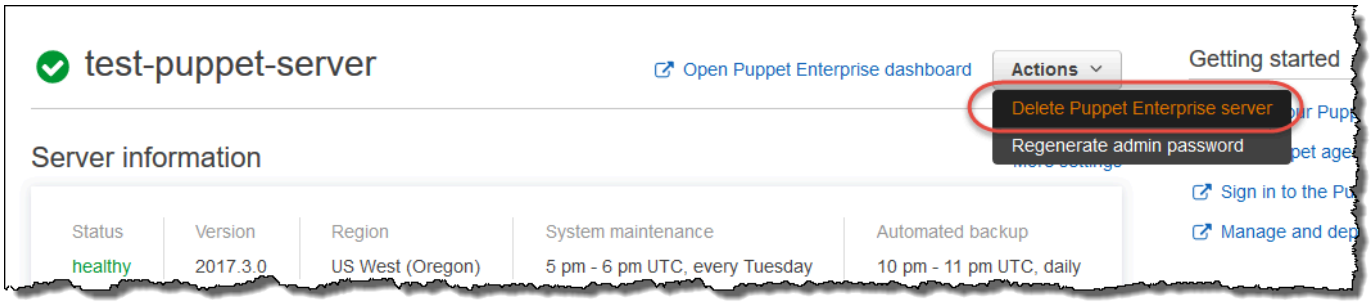
1. Dans le AWS CLI, exécutez la commande suivante pour dissocier les nœuds. *`Server_name`*est le nom du Puppet Master dont vous souhaitez dissocier le nœud. La valeur `--node-name` peut être un ID d'instance.

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

2. Attendez qu'un message de réponse indique que la dissociation est terminée.

Étape 2 : Suppression du serveur

1. Sur la vignette du serveur dans le tableau de bord, développez le menu Actions.



2. Choisissez Delete Puppet Enterprise server (Supprimer le serveur Puppet Enterprise).

3. Lorsque vous êtes invité à confirmer la suppression, activez la case pour supprimer les rôles et ressources associés, puis cliquez sur Oui, supprimer.

consultez aussi

- Dissocier un nœud d'un serveur OpsWorks for Puppet Enterprise

Comment migrer un serveur OpsWorks pour Puppet Enterprise vers Amazon Elastic Compute Cloud (Amazon EC2)

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les instructions ci-dessous décrivent comment migrer les serveurs Puppet Enterprise existants vers Amazon EC2, au cas où vous souhaiteriez continuer à utiliser Puppet Enterprise pour vos besoins de gestion de configuration en dehors de. OpsWorks

Rubriques

- Étape 1 : contactez Puppet pour acheter une licence
- Étape 2 : Obtenez des informations sur votre serveur OpsWorks pour Puppet Enterprise
- Étape 3 : effectuez une sauvegarde de votre serveur OpsWorks pour Puppet Enterprise
- Étape 4 : Lancer une nouvelle EC2 instance
- Étape 5 : Installation de Puppet Enterprise sur la nouvelle instance EC2
- Étape 6 : restauration de la sauvegarde sur la nouvelle EC2 instance
- Étape 7 : Configuration de votre licence Puppet
- Étape 8 : migrez vos nœuds
- Étape 9 : Supprimer votre serveur OpsWorks pour Puppet Enterprise

Étape 1 : contactez Puppet pour acheter une licence

Lorsque vous migrez vos serveurs vers EC2, la nouvelle instance n'est pas fournie avec une licence Puppet. Pour acheter une clé de licence, suivez les instructions sur le site Web de Puppet.

Étape 2 : Obtenez des informations sur votre serveur OpsWorks pour Puppet Enterprise

Recherchez et enregistrez les valeurs pour votre serveur OpsWorks pour Puppet Enterprise.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

   Copiez le nom du compartiment Amazon S3 existant pour votre serveur OpsWorks pour Puppet Enterprise. Le nom du bucket est au format suivant : `aws-opsworks-cm-`*`server-name`*`-`*`random-string`*

2. Exécutez la aws opsworks-cm describe-servers commande pour obtenir la configuration de votre serveur OpsWorks for Puppet Enterprise.

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

   Stockez les valeurs pour `InstanceTypeKeyPair`,`SubnetIds`,, `SecurityGroupIdsInstanceProfileArn`, et `Endpoint` issues de la réponse.

3. Utilisez SSH pour vous connecter au serveur existant OpsWorks pour Puppet Enterprise. Vous pouvez utiliser le gestionnaire de session dans la EC2 console au lieu de SSH.

   Exécutez la commande suivante.

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

   La réponse fournit la version de Puppet Enterprise (par exemple, 2019.8.10). Enregistrez cette valeur.

Vous utiliserez SSH ou le gestionnaire de session pour l'étape suivante.

Étape 3 : effectuez une sauvegarde de votre serveur OpsWorks pour Puppet Enterprise

1. Exécutez les commandes suivantes pour effectuer une sauvegarde locale.

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

2. Exécutez la commande suivante pour enregistrer le nom de la sauvegarde.

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

3. Exécutez la commande suivante pour télécharger votre sauvegarde dans un compartiment S3. Remplacez *S3-Bucket* par la valeur de l'étape 1 dansÉtape 2 : Obtenez des informations sur votre serveur OpsWorks pour Puppet Enterprise.

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

   Stockez les S3_BUCKET valeurs PUPPET_BACKUP et. Vous allez importer ces valeurs dans la nouvelle EC2 instance.

Vous pouvez quitter la session SSH ou le gestionnaire de session.

Étape 4 : Lancer une nouvelle EC2 instance

Lancez une nouvelle EC2 instance depuis la EC2 console https://console.aws.amazon.com/ec2/en utilisant la même configuration que celle du OpsWorks serveur Puppet Enterprise.

Nom du paramètre	Valeur
SE	Amazon Linux 2
Type d'instance	La `InstanceType` valeur de l'étape 2 deÉtape 2 : Obtenez des informations sur votre serveur OpsWorks pour Puppet Enterprise.

Nom du paramètre	Valeur
Nom de la paire de clés	La `KeyPair` valeur de l'étape 2 deÉtape 2 : Obtenez des informations sur votre serveur OpsWorks pour Puppet Enterprise.
VPC	Le VPC de l'étape 2 `SubnetIds` de. Étape 2 : Obtenez des informations sur votre serveur OpsWorks pour Puppet Enterprise
Sous-réseau	À `SubnetIds` partir de l'étape 2 deÉtape 2 : Obtenez des informations sur votre serveur OpsWorks pour Puppet Enterprise.
Sélectionnez un groupe de sécurité existant -> Groupes de sécurité communs	À `SecurityGroupIds` partir de l'étape 2 deÉtape 2 : Obtenez des informations sur votre serveur OpsWorks pour Puppet Enterprise.
Stockage	Au moins 120 Go.
Profil d'instance IAM	À `InstanceProfileArn` partir de l'étape 2 deÉtape 2 : Obtenez des informations sur votre serveur OpsWorks pour Puppet Enterprise.

Si vous souhaitez créer et associer une adresse IP élastique à la nouvelle instance, copiez l'ID d'instance de la nouvelle instance et suivez les étapes décrites dans(Facultatif) Étape 4.1 : créer et associer une adresse IP élastique.

(Facultatif) Étape 4.1 : créer et associer une adresse IP élastique

En utilisant une adresse IP Elastic, vous pouvez contourner un problème de défaillance d'une instance ou d'un logiciel en remappant rapidement l'adresse à une autre instance de votre compte.

Pour créer et associer une adresse IP élastique

1. Connectez-vous à la EC2 console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/ec2/.
2. Choisissez Elastic IPs.

3. Choisissez Allocate Elastic IP address (Allouer l'adresse IP Elastic).
4. Sur la page Allouer une adresse IP élastique, choisissez Allocate. Cela crée une IPv4 adresse publique.
5. Copiez l' IPv4 adresse allouée.
6. Dans Actions, sélectionnez Associer une adresse IP élastique.
7. Pour Instance, entrez l'ID d'instance pour la nouvelle instance.
8. Choisissez Associer.

Étape 5 : Installation de Puppet Enterprise sur la nouvelle instance EC2

Utilisez SSH pour vous connecter à la nouvelle EC2 instance. Vous pouvez utiliser le gestionnaire de session dans la EC2 console au lieu de SSH.

```
# switch to sudo user
sudo -i

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

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

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

Vous pouvez garder votre session SSH ou Session Manager ouverte pour l'étape suivante.

Étape 6 : restauration de la sauvegarde sur la nouvelle EC2 instance

```
# 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
```

Vous pouvez accéder à la console Puppet pour l' EC2 instance restaurée à l'adresse https ://*Public IPv4 of the instance*. Vous trouverez le IPv4 DNS public sur la page de détails de l'instance dans la EC2 console. Les informations de connexion sont les mêmes que celles que vous utilisez pour accéder à votre serveur OpsWorks pour Puppet Enterprise.

Vous pouvez garder votre session SSH ou Session Manager ouverte pour l'étape suivante.

Étape 7 : Configuration de votre licence Puppet

Suivez les étapes indiquées sur le site Web de Puppet pour configurer votre licence.

Vous pouvez garder votre session SSH ou Session Manager ouverte pour l'étape suivante.

Étape 8 : migrez vos nœuds

Deux types de domaines sont pris en charge par les serveurs OpsWorks for Puppet Enterprise :

- BYODC (apportez votre propre domaine et certificat)
- OpsWorks point de terminaison

Étape 8.1 : Pour le BYODC (apportez votre propre domaine et certificat)

Pour ces nœuds, il vous suffit de faire pointer le domaine personnalisé de votre fournisseur DNS vers le IPv4 DNS public ou l' IPv4 adresse publique de la nouvelle EC2 instance.

Étape 8.2 : Pour le OpsWorks terminal

Pour un OpsWorks point de terminaison, la documentation de Puppet recommande de désinstaller l'agent Puppet sur le nœud, puis d'installer l'agent Puppet à l'aide du serveur Puppet Enterprise récemment restauré.

> **Note**
>
> Bien que Puppet ne dispose pas de procédure automatique pour déplacer un nœud d'agent, les membres de la communauté Puppet ont publié quelques modules sur le site Web de Puppet Forge pour effectuer la migration automatique des nœuds. Ces modules incluent le `pe_migrate`module et un second module de migration créé par un autre auteur. Les modules du site Web de Puppet Forge ne sont pas pris en charge par Puppet ou OpsWorks sauf indication contraire explicite dans le module Forge. Nous vous recommandons de faire preuve de prudence avec ces modules et de les tester avant de les utiliser à grande échelle.

Les sections suivantes décrivent les étapes à suivre pour désinstaller et réinstaller les agents Puppet sur les instances Linux.

Rubriques

- Étape 8.2.1 : Copiez le programme de désinstallation depuis le serveur Puppet
- Étape 8.2.2 : Téléchargez le programme de désinstallation et exécutez-le sur un nœud
- Étape 8.2.3 : Réinstaller l'agent Puppet sur un nœud

Étape 8.2.1 : Copiez le programme de désinstallation depuis le serveur Puppet

Avant de désinstaller l'agent, assurez-vous que le profil d'instance IAM du nœud fournit ReadOnly les autorisations S3.

Exécutez la commande suivante pour copier le programme de désinstallation du serveur Puppet vers le compartiment S3.

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

Après avoir exécuté la commande, vous pouvez vous déconnecter de la session SSH ou du gestionnaire de session du serveur Puppet.

Étape 8.2.2 : Téléchargez le programme de désinstallation et exécutez-le sur un nœud

Utilisez SSH pour vous connecter au nœud. Vous pouvez utiliser le gestionnaire de session dans la EC2 console au lieu de SSH si le nœud est une EC2 instance.

```
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
```

Vous pouvez garder votre session SSH ou Session Manager ouverte pour l'étape suivante.

Étape 8.2.3 : Réinstaller l'agent Puppet sur un nœud

Procédez comme suit pour réinstaller l'agent Puppet sur un nœud.

Rubriques

- Étape 8.2.3.1 : Installation de l'agent Puppet avec la bonne configuration
- Étape 8.2.3.2 : Accepter le certificat dans la console Puppet
- Étape 8.2.3.3 : Vérifiez le nœud dans le serveur Puppet Enterprise

Étape 8.2.3.1 : Installation de l'agent Puppet avec la bonne configuration

Exécutez la commande suivante pour installer l'agent Puppet.

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

Vous pouvez garder votre session SSH ou Session Manager ouverte pour l'étape 8.2.2.3.

Étape 8.2.3.2 : Accepter le certificat dans la console Puppet

1. Accédez à la console du serveur Puppet à l'adresse`https://Public_IPv4_DNS`.
2. Choisissez Certificats, puis Certificats non signés.
3. Choisissez Accepter pour signer le certificat de l'agent Puppet.

Étape 8.2.3.3 : Vérifiez le nœud dans le serveur Puppet Enterprise

Exécutez la commande suivante sur le nœud pour l'intégrer au serveur.

```
puppet agent -t
```

Le nœud devrait maintenant être visible dans la console du serveur Puppet.

Étape 9 : Supprimer votre serveur OpsWorks pour Puppet Enterprise

Vous pouvez utiliser la OpsWorks console ou AWS CLI supprimer votre serveur OpsWorks pour Puppet Enterprise.

Pour supprimer votre serveur à l'aide de la OpsWorks console

1. Connectez-vous à la AWS OpsWorks console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/opsworks/.
2. Choisissez les serveurs Puppet Enterprise dans le volet de navigation.
3. Sur la page des serveurs Puppet Enterprise, choisissez le serveur que vous souhaitez supprimer.
4. Dans Actions, choisissez Supprimer le serveur Puppet Enterprise.

Pour supprimer votre serveur à l'aide du AWS CLI

Exécutez la commande suivante.

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

Journalisation OpsWorks des appels d'API Puppet Enterprise avec AWS CloudTrail

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que

possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

OpsWorks for Puppet Enterprise est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans OpsWorks Puppet Enterprise. CloudTrail capture tous les appels d'API OpsWorks pour Puppet Enterprise sous forme d'événements, y compris les appels depuis la console OpsWorks pour Puppet Enterprise et les appels de code vers Puppet Enterprise. OpsWorks APIs Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris des événements OpsWorks pour Puppet Enterprise. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande adressée à OpsWorks Puppet Enterprise, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite et des informations supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

OpsWorks pour obtenir des informations sur Puppet Enterprise dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité se produit dans OpsWorks Puppet Enterprise, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre AWS compte. Pour plus d'informations, consultez la section Affichage des événements avec l'historique des CloudTrail événements.

Pour un enregistrement continu des événements enregistrés sur votre AWS compte, y compris des événements OpsWorks pour Puppet Enterprise, créez un parcours. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez :

- Présentation de la création d'un journal d'activité
- CloudTrail Services et intégrations pris en charge

- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

Toutes les actions OpsWorks relatives à Puppet Enterprise sont enregistrées CloudTrail et documentées dans le manuel de référence de l'API OpsWorks pour Puppet Enterprise. Par exemple, les appels aux CreateServerCreateBackup, et DescribeServersles actions génèrent des entrées dans les fichiers CloudTrail journaux.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec des informations d'identification d'utilisateur root ou IAM.
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour plus d'informations, consultez la section Élément userIdentity CloudTrail .

Compréhension OpsWorks des entrées du fichier journal de Puppet Enterprise

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal pour l'`CreateServer`action 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"
}
```


Résolution des problèmes OpsWorks pour Puppet Enterprise

> **Important**
>
> Le AWS OpsWorks for Puppet Enterprise service a atteint sa fin de vie le 31 mars 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette rubrique contient certains problèmes courants liés OpsWorks à Puppet Enterprise, ainsi que des suggestions de solutions à ces problèmes.

Rubriques

- Conseils généraux de résolution des problèmes
- Résolution d'erreurs spécifiques

- Aide supplémentaire et support

Conseils généraux de résolution des problèmes

Si vous ne parvenez pas à créer ou à utiliser un maître Puppet, vous pouvez consulter les messages d'erreur ou les journaux pour vous aider à résoudre le problème. Les tâches suivantes expliquent par où vous devez généralement commencer pour résoudre un problème lié à un maître Puppet. Pour plus d'informations sur des erreurs spécifiques et les solutions, consultez la section Résolution d'erreurs spécifiques de cette rubrique.

- Utilisez la console OpsWorks for Puppet Enterprise pour afficher les messages d'erreur si un Puppet master ne démarre pas. Sur la page de propriétés du maître Puppet, les messages d'erreur liés au lancement et au fonctionnement du serveur sont affichés en haut de la page. Les erreurs peuvent provenir OpsWorks des services Puppet Enterprise ou Amazon EC2 utilisés pour créer un Puppet Master. AWS CloudFormation Sur la page de propriétés, vous pouvez aussi voir les événements qui se produisent sur un serveur en cours d'exécution et ainsi consulter les éventuels messages d'événement d'erreur.
- Pour résoudre EC2 les problèmes, connectez-vous à l'instance de votre serveur à l'aide du protocole SSH et consultez les journaux. EC2 les journaux d'instance sont stockés dans le `/var/log/aws/opsworks-cm` répertoire. Ces journaux capturent les sorties de commande tandis que OpsWorks pour Puppet Enterprise lance un Puppet Master.

Résolution d'erreurs spécifiques

Rubriques

- Le serveur est dans un état de perte de connexion
- La création du serveur échoue avec un message indiquant que la configuration demandée n'est actuellement pas prise en charge.
- Impossible de créer l' EC2 instance Amazon du serveur
- Une erreur de rôle de service empêche la création du serveur
- Limite appliquée aux adresses IP Elastic dépassée
- Une association de nœud sans surveillance échoue
- Échec de la maintenance du système

Le serveur est dans un état de perte de connexion

Problème : l'état d'un serveur indique que la connexion est perdue.

Cause : Cela se produit le plus souvent lorsqu'une entité extérieure AWS OpsWorks apporte des modifications à un serveur OpsWorks pour Puppet Enterprise ou à ses ressources de support. AWS OpsWorks Impossible de se connecter aux serveurs Puppet Enterprise en état de perte de connexion pour effectuer des tâches de maintenance telles que la création de sauvegardes, l'application de correctifs du système d'exploitation ou la mise à jour de Puppet. Par conséquent, il est possible que votre serveur ne dispose pas de mises à jour importantes, soit exposé à des problèmes de sécurité ou ne fonctionne pas comme prévu.

Solution : essayez les étapes suivantes pour rétablir la connexion du serveur.

1. Assurez-vous que votre rôle de service dispose de toutes les autorisations requises.

    a. Sur la page Paramètres de votre serveur, dans Réseau et sécurité, choisissez le lien correspondant au rôle de service utilisé par le serveur. Cela ouvre le rôle de service à afficher dans la console IAM.

    b. Dans l'onglet Autorisations, vérifiez que cela `AWSOpsWorksCMServiceRole` figure dans la liste des politiques d'autorisations. Si elle n'est pas répertoriée, ajoutez la politique `AWSOpsWorksCMServiceRole` gérée manuellement au rôle.

    c. Dans l'onglet Relations de confiance, vérifiez que le rôle de service dispose d'une politique de confiance qui autorise le `opsworks-cm.amazonaws.com` service à assumer des rôles en votre nom. Pour plus d'informations sur l'utilisation des politiques de confiance avec les rôles, consultez Modifier un rôle (console) ou le billet du blog sur la AWS sécurité intitulé Comment utiliser les politiques de confiance avec les rôles IAM.

2. Assurez-vous que votre profil d'instance dispose de toutes les autorisations requises.

    a. Sur la page Paramètres de votre serveur, dans Réseau et sécurité, choisissez le lien correspondant au profil d'instance utilisé par le serveur. Cela ouvre le profil de l'instance pour qu'il soit affiché dans la console IAM.

    b. Dans l'onglet Autorisations, vérifiez cela `AmazonEC2RoleforSSM` et `AWSOpsWorksCMInstanceProfileRole` les deux figurent dans la liste des politiques d'autorisations. Si l'une ou les deux ne figurent pas dans la liste, ajoutez ces politiques gérées manuellement au rôle.

    c. Dans l'onglet Relations de confiance, vérifiez que le rôle de service dispose d'une politique de confiance qui autorise le `ec2.amazonaws.com` service à assumer des rôles en votre

nom. Pour plus d'informations sur l'utilisation des politiques de confiance avec les rôles, consultez Modifier un rôle (console) ou le billet du blog sur la AWS sécurité intitulé Comment utiliser les politiques de confiance avec les rôles IAM.

3. Dans la EC2 console Amazon, assurez-vous que vous vous trouvez dans la même région que celle du serveur OpsWorks for Puppet Enterprise, puis redémarrez l' EC2 instance utilisée par votre serveur.

   a. Choisissez l' EC2 instance nommée `aws-opsworks-cm-instance-`*`server-name`*.

   b. Dans le menu État de l'instance, choisissez Redémarrer l'instance.

   c. Prévoyez jusqu'à 15 minutes pour que votre serveur redémarre et soit entièrement en ligne.

4. Dans la console OpsWorks for Puppet Enterprise, sur la page des détails du serveur, vérifiez que l'état du serveur est désormais sain.

Si l'état du serveur est toujours Perte de connexion après avoir effectué les étapes précédentes, essayez l'une des solutions suivantes.

- Remplacez le serveur en créant un nouveau et en supprimant l'original. Si les données du serveur actuel sont importantes pour vous, restaurez le serveur à partir d'une sauvegarde récente et vérifiez que les données sont à jour avant de supprimer le serveur d'origine qui ne répond pas.
- Contactez AWS le support.

La création du serveur échoue avec un message indiquant que la configuration demandée n'est actuellement pas prise en charge.

Problème : Vous essayez de créer un serveur Puppet Enterprise, mais l'opération échoue avec un message d'erreur de type « The requested configuration is currently not supported. Please check the documentation for supported configurations. »

Cause : Un type d'instance non pris en charge a peut-être été spécifié pour le maître Puppet. Si vous choisissez de créer le serveur Puppet dans un VPC disposant d'une location autre que celle par défaut, comme une location pour des instances dédiées, toutes les instances à l'intérieur du VPC spécifié doivent être également associées à une location dédiée ou d'hôte. Certains types d'instance, comme t2, ne sont disponibles qu'avec la location par défaut. Il est donc possible que le type d'instance du maître Puppet ne soit pas pris en charge par le VPC spécifié, c'est pourquoi la création du serveur échoue.

Solution : Si vous choisissez un VPC disposant d'une location autre que celle par défaut, utilisez un type d'instance m4 qui peut prendre en charge une location dédiée.

Impossible de créer l' EC2 instance Amazon du serveur

Problème : La création du serveur a échoué avec un message d'erreur similaire au suivant : « Impossible de créer les ressources suivantes : [EC2Instance]. Failed to receive 1 resource signal(s) within the specified duration. »

Cause : Cela est probablement dû au fait que l' EC2instance n'a pas accès au réseau.

Solution : assurez-vous que l'instance dispose d'un accès Internet sortant et que l'agent de AWS service est en mesure d'émettre des commandes. Assurez-vous que le paramètre Résolution DNS est activé pour votre VPC (un VPC avec un seul sous-réseau public), et que le paramètre Activer l'adresse IP publique attribuée automatiquement est activé pour votre sous-réseau.

Une erreur de rôle de service empêche la création du serveur

Problème : la création du serveur échoue avec un message d'erreur indiquant « Non autorisé à exécuter sts : »AssumeRole.

Cause : Cela peut se produire lorsque le rôle de service vous utilisez n'a pas les autorisations appropriées pour créer un nouveau serveur.

Solution : Ouvrez la console OpsWorks for Puppet Enterprise ; utilisez-la pour générer un nouveau rôle de service et un rôle de profil d'instance. Si vous préférez utiliser votre propre rôle de service, associez la politique AWSOpsWorks CMService Role au rôle. Vérifiez que opsworks-cm.amazonaws.com figure parmi les services concernés par les relations de confiance du rôle. Vérifiez que le rôle de service associé au Puppet master est associé à la politique de gestion AWSOpsdes CMService rôles de travail attachée.

Limite appliquée aux adresses IP Elastic dépassée

Problème : la création du serveur échoue avec un message d'erreur indiquant : « Impossible de créer les ressources suivantes : [EIP, EC2 instance]. Resource creation cancelled, the maximum number of addresses has been reached. »

Cause : Cette erreur a lieu lorsque votre compte a utilisé le nombre maximal d'adresses IP Elastic (EIP). La limite d'adresses IP Elastic est de cinq.

Solution : vous pouvez soit libérer les adresses EIP existantes, soit supprimer celles que votre compte n'utilise pas activement, soit contacter le Support AWS client pour augmenter le nombre d'adresses EIP associées à votre compte.

Une association de nœud sans surveillance échoue

Problème : l'association automatique ou non surveillée de nouveaux EC2 nœuds Amazon échoue. Les nœuds qui auraient dû être ajoutés au maître Puppet ne sont pas présents sur le tableau de bord Puppet Enterprise.

Cause : Cela peut se produire lorsqu'aucun rôle IAM n'est configuré en tant que profil d'instance permettant aux appels d'`opsworks-cm`API de communiquer avec de nouvelles EC2 instances.

Solution : associez à votre profil d' EC2 instance une politique permettant aux appels `AssociateNode` et à `DescribeNodeAssociationStatus` l'API de fonctionner EC2, comme décrit dansAjout automatique de nœuds dans OpsWorks Puppet Enterprise.

Échec de la maintenance du système

AWS OpsWorks CM effectue une maintenance hebdomadaire du système afin de s'assurer que les dernières versions AWS testées de Puppet Server, y compris les mises à jour de sécurité, fonctionnent toujours sur un serveur OpsWorks pour Puppet Enterprise. Si, pour une raison quelconque, la maintenance du système échoue, vous en AWS OpsWorks CM informe. Pour plus d'informations sur la maintenance du système, consultezMaintenance du système dans OpsWorks Puppet Enterprise.

Cette section décrit les causes possibles de l'échec et propose des solutions.

Rubriques

- Une erreur de rôle de service ou de profil d'instance empêche la maintenance du système

Une erreur de rôle de service ou de profil d'instance empêche la maintenance du système

Problème : la maintenance du système échoue avec un message d'erreur indiquant « Non autorisé à exécuter sts : AssumeRole » ou un message d'erreur similaire concernant les autorisations.

Cause : Cela peut se produire lorsque le rôle de service ou le profil d'instance que vous utilisez ne dispose pas des autorisations adéquates pour effectuer la maintenance du système sur le serveur.

Solution : Assurez-vous que votre rôle de service et votre profil d'instance disposent de toutes les autorisations requises.

1. Assurez-vous que votre rôle de service dispose de toutes les autorisations requises.

   a. Sur la page Paramètres de votre serveur, dans Réseau et sécurité, choisissez le lien correspondant au rôle de service utilisé par le serveur. Cela ouvre le rôle de service à afficher dans la console IAM.

   b. Dans l'onglet Autorisations, vérifiez qu'`AWSOpsWorksCMServiceRole`il est attaché au rôle de service. Si elle n'`AWSOpsWorksCMServiceRole`est pas répertoriée, ajoutez cette politique au rôle.

   c. Vérifiez que opsworks-cm.amazonaws.com figure parmi les services concernés par les relations de confiance du rôle. Pour plus d'informations sur l'utilisation des politiques de confiance avec les rôles, consultez Modifier un rôle (console) ou le billet du blog sur la AWS sécurité intitulé Comment utiliser les politiques de confiance avec les rôles IAM.

2. Assurez-vous que votre profil d'instance dispose de toutes les autorisations requises.

   a. Sur la page Paramètres de votre serveur, dans Réseau et sécurité, choisissez le lien correspondant au profil d'instance utilisé par le serveur. Cela ouvre le profil de l'instance pour qu'il soit affiché dans la console IAM.

   b. Dans l'onglet Autorisations, vérifiez cela `AmazonEC2RoleforSSM` et `AWSOpsWorksCMInstanceProfileRole` les deux figurent dans la liste des politiques d'autorisations. Si l'une ou les deux ne figurent pas dans la liste, ajoutez ces politiques gérées manuellement au rôle.

   c. Dans l'onglet Relations de confiance, vérifiez que le rôle de service dispose d'une politique de confiance qui autorise le `ec2.amazonaws.com` service à assumer des rôles en votre nom. Pour plus d'informations sur l'utilisation des politiques de confiance avec les rôles, consultez Modifier un rôle (console) ou le billet du blog sur la AWS sécurité intitulé Comment utiliser les politiques de confiance avec les rôles IAM.

Aide supplémentaire et support

Si vous ne voyez pas votre problème spécifique décrit dans cette rubrique, ou si vous avez essayé les suggestions de cette rubrique et que les problèmes persistent, consultez les forums AWS OpsWorks.

Vous pouvez également visiter le Centre de Support AWS. Le Centre de AWS support est le centre de création et de gestion des dossiers de AWS support. Le Centre de AWS support inclut également

des liens vers d'autres ressources utiles, telles que des forums, des informations techniques FAQs, l'état de santé des services et AWS Trusted Advisor.

AWS OpsWorks pour Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks for Chef Automate vous permet d'exécuter un serveur Chef Automate dans AWS. Vous pouvez configurer un serveur Chef en quelques minutes et le laisser AWS OpsWorks for Chef Automate gérer ses opérations, ses sauvegardes, ses restaurations et ses mises à niveau logicielles. AWS OpsWorks for Chef Automate vous permet de vous concentrer sur les tâches de gestion de configuration de base, au lieu de gérer un serveur Chef.

Un serveur Chef Automate gère la configuration des nœuds de votre environnement en indiquant les recettes Chef à exécuter sur les nœuds, en stockant `chef-client`des informations sur les nœuds et en servant de référentiel central pour vos livres de recettes Chef. AWS OpsWorks for Chef Automate fournit des serveurs Chef qui incluent des fonctionnalités premium de Chef Automate : Chef Infra et Chef InSpec.

Un AWS OpsWorks for Chef Automate serveur s'exécute sur une instance Amazon Elastic Compute Cloud. AWS OpsWorks for Chef Automate les serveurs sont configurés pour exécuter la dernière version d'Amazon Linux (Amazon Linux 2). Pour plus d'informations sur ce qui a changé dans cette version de Chef Automate, consultez les Notes de mise à jour de Chef Automate. Le tableau suivant décrit les composants Chef installés sur un AWS OpsWorks for Chef Automate serveur.

Nom du composant	Description	Version installée sur le AWS OpsWorks for Chef Automate serveur
Chef Automate	Chef Automate est un package logiciel de serveur d'entreprise qui fournit un flux de travail automatisé pour un déploiement continu, et des	2.0

Nom du composant	Description	Version installée sur le AWS OpsWorks for Chef Automate serveur
	informations sur les nœuds gérés dans une console de gestion web. Chef Automate assure l'automatisation de l'infrastructure en incluant Chef Infra, les informations de sécurité et de conformité et leur application en incluant Chef InSpec, et le déploiement automatisé en incluant Chef Habitat. Pour plus d'informations sur Chef Automate, consultez Chef Automate sur le site web Chef.	
Chef Infra	Anciennement nommé Chef Server, Chef Infra Server utilise l'agent du client Chef Infra (`chef-client`) pour appliquer en permanence les configurations aux nœuds gérés afin de maintenir un état souhaité. Pour plus d'informations sur Infra, consultez Chef Infra sur le site web Chef.	12.x

Nom du composant	Description	Version installée sur le AWS OpsWorks for Chef Automate serveur
Chef InSpec	Chef InSpec décrit les règles de sécurité et de conformité qui peuvent être partagées entre les ingénieurs logiciels, les ingénieurs des opérations et les ingénieurs de sécurité. Les exigences de conformit é, de sécurité et les autres exigences en matière de stratégie constituent le cadre des tests automatiques que l'agent `chef-client` peut exécuter sur les nœuds gérés, en garantissant une application cohérente des normes. Pour plus d'informations InSpec, consultez Chef InSpec sur le site Web de Chef.	3.9.0

La version minimale de `chef-client` prise en charge sur les nœuds associés à un serveur AWS OpsWorks for Chef Automate est 13.x. Nous recommandons d'exécuter au moins la version 14.10.9, ou la version stable `chef-client` la plus récente.

Lorsque de nouvelles versions mineures du logiciel Chef deviennent disponibles, la maintenance système est conçue pour mettre à jour automatiquement la version mineure de Chef Automate et de Chef Server sur le serveur dès que celle-ci réussit les tests AWS. AWS effectue des tests approfondis pour vérifier que les mises à niveau de Chef sont prêtes pour la production et ne perturbent pas les environnements clients existants. Il peut donc y avoir des retards entre les versions du logiciel Chef et leur disponibilité en tant qu'application par rapport aux serveurs existants

OpsWorks pour les serveurs Chef Automate. Cette clé autorise le serveur à s'identifier auprès de l'agent chef-client sur les nœuds gérés par votre serveur.

Vous pouvez connecter n'importe quel ordinateur ou EC2 instance sur site exécutant un système d'exploitation compatible et disposant d'un accès réseau à un AWS OpsWorks for Chef Automate serveur. Pour obtenir une liste des systèmes d'exploitation pour les nœuds que vous voulez gérer, consultez le site web Chef. Le logiciel agent `chef-client` est installé sur les nœuds que vous voulez gérer avec un serveur Chef.

Rubriques

- Support régional pour AWS OpsWorks for Chef Automate
- AWS OpsWorks pour Chef Automate End of Life FAQs
- Mettre à niveau un AWS OpsWorks for Chef Automate serveur vers Chef Automate 2
- Commencer avec AWS OpsWorks for Chef Automate
- Créez un AWS OpsWorks for Chef Automate serveur en utilisant AWS CloudFormation
- Mettre à jour un AWS OpsWorks for Chef Automate serveur pour utiliser un domaine personnalisé
- Régénérer le kit de démarrage d'un serveur AWS OpsWorks for Chef Automate
- Utilisation des balises sur les AWS OpsWorks for Chef Automate ressources
- Sauvegarder et restaurer un AWS OpsWorks for Chef Automate serveur
- Maintenance du système dans AWS OpsWorks for Chef Automate
- Analyses de conformité dans AWS OpsWorks for Chef Automate
- Dissocier un nœud d'un serveur AWS OpsWorks for Chef Automate
- Supprimer un AWS OpsWorks for Chef Automate serveur
- Réinitialiser les informations d'identification du tableau de bord Chef Automate
- Journalisation des appels d' AWS OpsWorks for Chef Automate API avec AWS CloudTrail
- Résolution des problèmes AWS OpsWorks for Chef Automate

Support régional pour AWS OpsWorks for Chef Automate

Les points de terminaison régionaux suivants prennent en charge AWS OpsWorks for Chef Automate les serveurs. AWS OpsWorks for Chef Automate crée des ressources associées à vos serveurs Chef, telles que des profils d'instance, des utilisateurs et des rôles de service, dans le même point de

terminaison régional que votre serveur Chef. Votre serveur Chef doit se trouver dans un VPC. Vous pouvez utiliser un VPC que vous créez ou que vous possédez déjà, ou utiliser le VPC par défaut.

- Région US East (Ohio)
- Région US East (N. Virginia)
- Région US West (N. California)
- Région US West (Oregon)
- Région Asie-Pacifique (Tokyo)
- Région Asia Pacific (Singapore)
- Région Asia Pacific (Sydney)
- Région Europe (Frankfurt)
- Europe (Ireland) Region

AWS OpsWorks pour Chef Automate End of Life FAQs

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative.

Rubriques

- Comment les utilisateurs existants seront-ils affectés par cette fin de vie ?
- Qu'arrive-t-il à mes serveurs si je n'agis pas ?
- Quelles sont les alternatives vers lesquelles je peux effectuer la transition ?
- Le service accepte-t-il toujours de nouveaux clients ?
- La fin de vie affectera-t-elle Régions AWS tout le monde en même temps ?
- Quel est le niveau de support technique disponible ?
- Je suis actuellement client de OpsWorks for Chef Automate et je dois lancer un serveur sur un compte qui n'utilisait pas le service auparavant. Suis-je capable de le faire ?
- Y aura-t-il des nouveautés majeures au cours de l'année prochaine ?

Comment les utilisateurs existants seront-ils affectés par cette fin de vie ?

Les clients existants ne seront pas affectés jusqu'au 5 mai 2024, date de fin de vie OpsWorks de Chef Automate. Après la date de fin de vie, les clients ne seront plus en mesure de gérer leurs serveurs à l'aide de la OpsWorks console ou de l'API.

Qu'arrive-t-il à mes serveurs si je n'agis pas ?

À compter du 5 mai 2024, vous ne pourrez plus gérer vos serveurs à l'aide de la OpsWorks console ou de l'API. À ce moment-là, nous cesserons d'exécuter les fonctions de gestion continues de vos serveurs, telles que les sauvegardes ou la maintenance. Pour limiter l'impact sur les clients, nous laisserons fonctionner toutes les EC2 instances qui sauvegardent les serveurs Chef Automate, mais leurs licences ne seront plus valides car leur utilisation n'est plus couverte (ou facturée) dans le cadre du contrat OpsWorks de service conclu avec Chef Automate. Vous devrez contacter Chef pour obtenir une nouvelle licence. Lorsque vous contactez Chef, assurez-vous de leur indiquer que vous êtes déjà un client OpsWorks de Chef Automate et que vous êtes en transition depuis OpsWorks.

Quelles sont les alternatives vers lesquelles je peux effectuer la transition ?

AWS et Progress Chef vous recommandent de migrer vers leur nouvelle offre Chef SaaS afin que vous puissiez continuer à bénéficier d'un service Chef Automate entièrement géré. Pour démarrer avec Chef SaaS, vous pouvez contacter Chef pour obtenir de la documentation sur la configuration d'un compte Chef SaaS et la transition de vos données et de vos nœuds.

Si Chef SaaS ne répond pas à vos besoins parce que vous préférez exécuter Chef Automate sur des EC2 instances de AWS comptes que vous contrôlez, Chef propose plusieurs options, notamment un modèle de licence AWS Marketplace Bring Your Own (BYOL) et l'auto-hébergement. EC2 Vous pouvez contacter Progress Chef pour plus d'informations sur la manière d'exécuter une telle transition.

Le service accepte-t-il toujours de nouveaux clients ?

Non. AWS OpsWorks for Chef Automate n'accepte plus de nouveaux clients.

La fin de vie affectera-t-elle Régions AWS tout le monde en même temps ?

Oui. L'API et la console atteindront leur fin de vie et seront totalement inutilisables à partir du 5 mai 2024. Régions AWS Pour plus d'informations sur Régions AWS la AWS OpsWorks disponibilité de Chef Automate, consultez la liste des services AWS régionaux.

Quel est le niveau de support technique disponible ?

AWS continuera à fournir le même niveau de support OpsWorks pour Chef Automate que celui dont bénéficient les clients aujourd'hui jusqu'à la date de fin de vie. Si vous avez des questions ou des préoccupations, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium. Pour obtenir de l'aide à la transition, nous recommandons aux clients de contacter Progress Chef.

Je suis actuellement client de OpsWorks for Chef Automate et je dois lancer un serveur sur un compte qui n'utilisait pas le service auparavant. Suis-je capable de le faire ?

En général non, sauf dans des circonstances exceptionnelles. Si vous vous trouvez dans une situation particulière, contactez l' AWS Support équipe sur AWS Re:post ou via le AWS Support Premium avec les détails et les justifications nécessaires, et nous examinerons votre demande.

Y aura-t-il des nouveautés majeures au cours de l'année prochaine ?

Non. Le service étant sur le point d'atteindre sa fin de vie, nous ne publierons aucune nouvelle fonctionnalité. Cependant, nous continuerons à améliorer la sécurité et à gérer les serveurs comme prévu jusqu'à la date de fin de vie.

Mettre à niveau un AWS OpsWorks for Chef Automate serveur vers Chef Automate 2

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Conditions préalables à la mise à niveau vers Chef Automate 2

Avant de commencer, assurez-vous de comprendre les nouvelles fonctions ajoutées par Chef Automate 2 ainsi que les fonctions que Chef Automate 2 ne prend pas en charge. Pour de plus

amples informations sur les nouvelles fonctions et les fonctions non prises en charge dans Chef Automate 2, consultez la documentation Chef Automate 2 sur le site web de Chef.

Pour qu'un serveur exécutant Chef Automate 1 soit éligible à la mise à niveau, au moins une maintenance doit avoir réussi après le 1er novembre 2019 sur ce serveur.

Comme pour toute opération de maintenance sur votre AWS OpsWorks for Chef Automate serveur, celui-ci est hors ligne pendant la mise à niveau. Vous devez prévoir un temps d'arrêt pouvant durer jusqu'à trois heures pendant le processus de mise à niveau.

Vous avez besoin des informations d'identification de connexion pour ce serveur pour le site web du tableau de bord Chef Automate. Lorsque la mise à niveau est terminée, vous devez vous connecter au tableau de bord Chef Automate et vérifier que vos nœuds et les informations de configuration ne sont pas modifiés.

> **Important**
>
> Lorsque vous êtes prêt à mettre à niveau votre AWS OpsWorks for Chef Automate serveur vers Chef Automate 2, utilisez uniquement les instructions ici pour effectuer la mise à niveau. Dans la mesure AWS OpsWorks for Chef Automate où de nombreux processus de mise à niveau sont automatisés, tels que la création de sauvegardes, ne suivez pas les instructions de mise à niveau sur le site Web de Chef.

À propos du processus de mise à niveau

Pendant le processus de mise à niveau, votre serveur est sauvegardé avant le démarrage de la mise à niveau et après la fin de la mise à niveau. Les sauvegardes suivantes sont créées :

- Une sauvegarde du serveur lorsqu'il exécute encore Chef Automate 1 (version 12.17.33)
- Une sauvegarde du serveur après la fin de la mise à niveau et lorsque le serveur exécute Chef Automate 2 (version 2019-08).

Le processus de mise à niveau met fin à l' EC2 instance Amazon que le serveur utilisait lorsqu'il a exécuté Chef Automate 1. Une nouvelle instance est créée pour exécuter le serveur Chef Automate 2.

Mise à niveau vers Chef Automate 2 (console)

1. Connectez-vous à la AWS OpsWorks console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/opsworks/.
2. Dans le panneau de navigation de gauche, choisissez AWS OpsWorks for Chef Automate.
3. Choisissez un serveur pour afficher la page de ses propriétés. Une bannière bleue en haut de la page doit indiquer si le serveur est éligible à la mise à niveau vers Chef Automate 2.

   > **Note**
   >
   > Pour qu'un serveur exécutant Chef Automate 1 soit éligible à la mise à niveau, au moins une maintenance doit avoir réussi après le 1er novembre 2019 sur ce serveur.

4. Si le serveur est éligible pour la mise à niveau, choisissez Start upgrade (Démarrer la mise à niveau).
5. Prévoyez jusqu'à trois heures pour la mise à niveau. Au cours du processus de mise à niveau, la page des propriétés affiche l'état Under maintenance (En cours de maintenance) pour le serveur.
6. Une fois la mise à niveau terminée, la page des propriétés affiche les deux messages suivants : Successfully upgraded to Automate 2 (Mise à niveau réussie vers Automate 2) et Maintenance completed successfully (Maintenance terminée avec succès). L'état du serveur doit être HEALTHY (SAIN).
7. Connectez-vous au tableau de bord Chef Automate avec vos informations d'identification existantes et vérifiez que vos nœuds sont correctement signalés.

Mise à niveau vers Chef Automate 2 (interface de ligne de commande)

1. (Facultatif) Si vous ne savez pas lequel de vos AWS OpsWorks for Chef Automate serveurs est éligible à la mise à niveau, exécutez la commande suivante. Assurez-vous d'ajouter le `--region` paramètre si vous souhaitez répertorier AWS OpsWorks for Chef Automate les serveurs d'une région AWS différente de votre région AWS par défaut.

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

   Dans les résultats, recherchez la valeur `true` pour l'attribut `CHEF_MAJOR_UPGRADE_AVAILABLE`. Cette valeur indique que le serveur peut être mis à niveau

vers Chef Automate 2. Notez les noms des AWS OpsWorks for Chef Automate serveurs éligibles à la mise à niveau.

2. Exécutez la commande suivante en *server_name* remplaçant par le nom d'un AWS OpsWorks for Chef Automate serveur. Pour effectuer une mise à niveau vers Chef Automate 2 au lieu d'effectuer une maintenance système de routine, ajoutez l'attribut de moteur `CHEF_MAJOR_UPGRADE`, comme indiqué dans la commande. Ajoutez le paramètre `--region` si le serveur cible n'est pas dans votre région AWS par défaut. Vous ne pouvez mettre à niveau qu'un serveur par commande.

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

   Si, pour une quelconque raison, le serveur AWS OpsWorks for Chef Automate ne peut pas être mis à niveau, cette commande entraîne une exception de validation.

3. Prévoyez jusqu'à trois heures pour la mise à niveau. Vous pouvez vérifier périodiquement l'état de la mise à niveau en exécutant la commande suivante.

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

   Dans les résultats, recherchez la valeur `Status`. Le code`Status` de `UNDER_MAINTENANCE` signifie que la mise à niveau est toujours en cours. Une mise à niveau réussie renvoie des messages similaires aux messages suivants.

```
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
```

   Si la mise à niveau échoue, AWS OpsWorks for Chef Automate restaurez automatiquement votre serveur vers Chef Automate 1.

   Si la mise à niveau a réussi mais que le serveur ne fonctionne pas de la même manière qu'avant la mise à niveau (par exemple, si les nœuds gérés ne sont pas signalés), vous pouvez restaurer manuellement le serveur. Pour de plus amples informations sur la restauration manuelle, veuillez consulter Restaurer un AWS OpsWorks for Chef Automate serveur vers Chef Automate 1 (CLI).

Restaurer un AWS OpsWorks for Chef Automate serveur vers Chef Automate 1 (CLI)

Si le processus de mise à niveau échoue, ramène AWS OpsWorks for Chef Automate automatiquement votre serveur vers Chef Automate 1. Si la mise à niveau a réussi mais que le serveur ne fonctionne pas de la même manière qu'avant la mise à niveau, vous pouvez rétablir manuellement votre AWS OpsWorks for Chef Automate serveur vers Chef Automate 1 en utilisant le AWS CLI.

1. Exécutez la commande suivante pour afficher le `BackupId` de la dernière sauvegarde effectuée sur votre serveur avant la tentative de mise à niveau. Ajoutez le paramètre `--region` si votre serveur se trouve dans une région AWS différente de votre région AWS par défaut.

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

   IDs Les sauvegardes sont au format*ServerName-yyyyMMddHHmmssSSS*. Recherchez les propriétés de Chef Automate 1 suivantes dans les résultats.

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

2. Exécutez la commande suivante en utilisant l'ID de sauvegarde renvoyée à l'étape 1 comme valeur de `--backup-id`.

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

   La restauration du serveur peut prendre entre 20 minutes et trois heures, en fonction de la quantité de données que vous avez stockées sur le serveur. Pendant l'opération de restauration, l'état de votre serveur est `RESTORING`. Cet état est affiché sur la page des propriétés du serveur dans le AWS Management Console, et renvoyé dans les résultats de la describe-servers commande.

3. Une fois la restauration terminée, la console affiche le message Restore completed successfully (Restauration terminée avec succès). Votre AWS OpsWorks for Chef Automate serveur est en ligne, comme avant que vous ne commenciez le processus de mise à niveau.

consultez aussi

- Maintenance du système dans AWS OpsWorks for Chef Automate
- Restaurer un AWS OpsWorks for Chef Automate serveur à partir d'une sauvegarde
- DescribeServers dans la Référence d'API AWS OpsWorks
- StartMaintenance dans la Référence d'API AWS OpsWorks

Commencer avec AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks for Chef Automate vous permet d'exécuter un serveur Chef Automate dans AWS. Vous pouvez mettre en service un serveur Chef en 15 minutes environ.

À partir du 3 mai 2021, AWS OpsWorks for Chef Automate stocke certains attributs du serveur Chef Automate dans AWS Secrets Manager. Pour de plus amples informations, veuillez consulter Intégration avec AWS Secrets Manager.

La procédure pas à pas suivante vous aide à créer votre premier serveur Chef dans AWS OpsWorks for Chef Automate.

Prérequis

Avant de commencer, vous devez remplir les conditions préalables suivantes.

Rubriques

- Configuration d'un VPC
- Prérequis pour l'utilisation d'un domaine personnalisé (facultatif)
- Configurer une paire de EC2 clés (facultatif)

Configuration d'un VPC

Votre AWS OpsWorks for Chef Automate serveur doit fonctionner dans un Amazon Virtual Private Cloud. Vous pouvez l'ajouter à un VPC existant, utiliser le VPC par défaut, ou créer un VPC pour contenir le serveur. Pour plus d'informations sur Amazon VPC et sur la façon de créer un nouveau VPC, consultez le guide de démarrage Amazon VPC.

Si vous créez votre propre VPC ou si vous utilisez un VPC existant, celui-ci doit avoir les propriétés ou paramètres suivants.

- Le VPC doit avoir au moins un sous-réseau.

  Si votre AWS OpsWorks for Chef Automate serveur doit être accessible au public, rendez le sous-réseau public et activez l'attribution automatique d'une adresse IP publique.
- L'option Résolution DNS doit être activée.
- Sur le sous-réseau, activez Attribuer automatiquement l'adresse IP publique.

Prérequis pour l'utilisation d'un domaine personnalisé (facultatif)

Vous pouvez configurer votre serveur Chef Automate sur votre propre domaine, en spécifiant un point de terminaison public dans un domaine personnalisé à utiliser comme point de terminaison de votre serveur. Lorsque vous utilisez un domaine personnalisé, tous les éléments et opérations ci-après sont requis, comme décrit en détail dans cette section.

Rubriques

- Configurer un domaine personnalisé
- Obtenir un certificat
- Obtenir une clé privée

Configurer un domaine personnalisé

Pour exécuter votre serveur Chef Automate sur votre propre domaine personnalisé, vous avez besoin d'un point de terminaison public d'un serveur, tel que `https://aws.my-company.com`. Si vous spécifiez un domaine personnalisé, vous devez également fournir un certificat et une clé privée, comme décrit dans les sections précédentes.

Pour accéder au serveur après sa création, ajoutez un enregistrement DNS CNAME dans votre service DNS préféré. Cet enregistrement doit faire pointer le domaine personnalisé vers le point de

terminaison (la valeur de l'attribut `Endpoint` du serveur) généré par le processus de création du serveur Chef Automate. Vous ne pouvez pas accéder au serveur à l'aide de la valeur `Endpoint` générée si le serveur utilise un domaine personnalisé.

Obtenir un certificat

Pour configurer votre serveur Chef Automate sur votre propre domaine personnalisé, vous avez besoin d'un certificat HTTPS au format PEM. Il peut s'agir d'un seul certificat auto-signé ou d'une chaîne de certificats. Lorsque vous effectuez le flux de travail Create Chef Automate server (Créer un serveur Chef Automate), si vous spécifiez ce certificat, vous devez également fournir un domaine et une clé privée personnalisés.

Voici les conditions requises pour la valeur du certificat :

- Vous pouvez fournir un certificat personnalisé auto-signé ou la chaîne de certificats complète.
- Le certificat doit être un certificat X509 valide ou une chaîne de certificats au format PEM.
- Le certificat doit être valide au moment du chargement. Vous ne pouvez pas importer un certificat avant le début de sa période de validité (date `NotBefore` du certificat) ou après son expiration (date `NotAfter` du certificat).
- Le nom commun ou les noms alternatifs du sujet (SANs) du certificat, s'ils sont présents, doivent correspondre à la valeur de domaine personnalisée.
- Le certificat doit correspondre à la valeur du champ Custom private key (Clé privée personnalisée) .

Obtenir une clé privée

Pour configurer votre serveur Chef Automate sur votre propre domaine personnalisé, vous avez besoin d'une clé privée au format PEM pour vous connecter au serveur à l'aide du protocole HTTPS. La clé privée ne doit pas être chiffrée ; elle ne peut pas être protégée par un mot de passe ou une phrase passe. Si vous spécifiez une clé privée personnalisée, vous devez également fournir un domaine et un certificat personnalisés.

Configurer une paire de EC2 clés (facultatif)

Aucune connexion SSH n'est nécessaire ni recommandée pour une gestion classique du serveur Chef ; vous pouvez utiliser les commandes `knife` pour exécuter la plupart des tâches de gestion sur votre serveur Chef.

Une paire de EC2 clés est requise pour vous connecter à votre serveur en utilisant SSH au cas où vous perdriez ou souhaiteriez modifier le mot de passe de connexion au tableau de bord Chef Automate. Vous pouvez utiliser une paire de clés existante, ou en créer une. Pour plus d'informations sur la création d'une nouvelle paire de EC2 clés, consultez Amazon EC2 Key Pairs.

Si vous n'avez pas besoin d'une paire de EC2 clés, vous êtes prêt à créer un serveur Chef.

Création d'un serveur Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

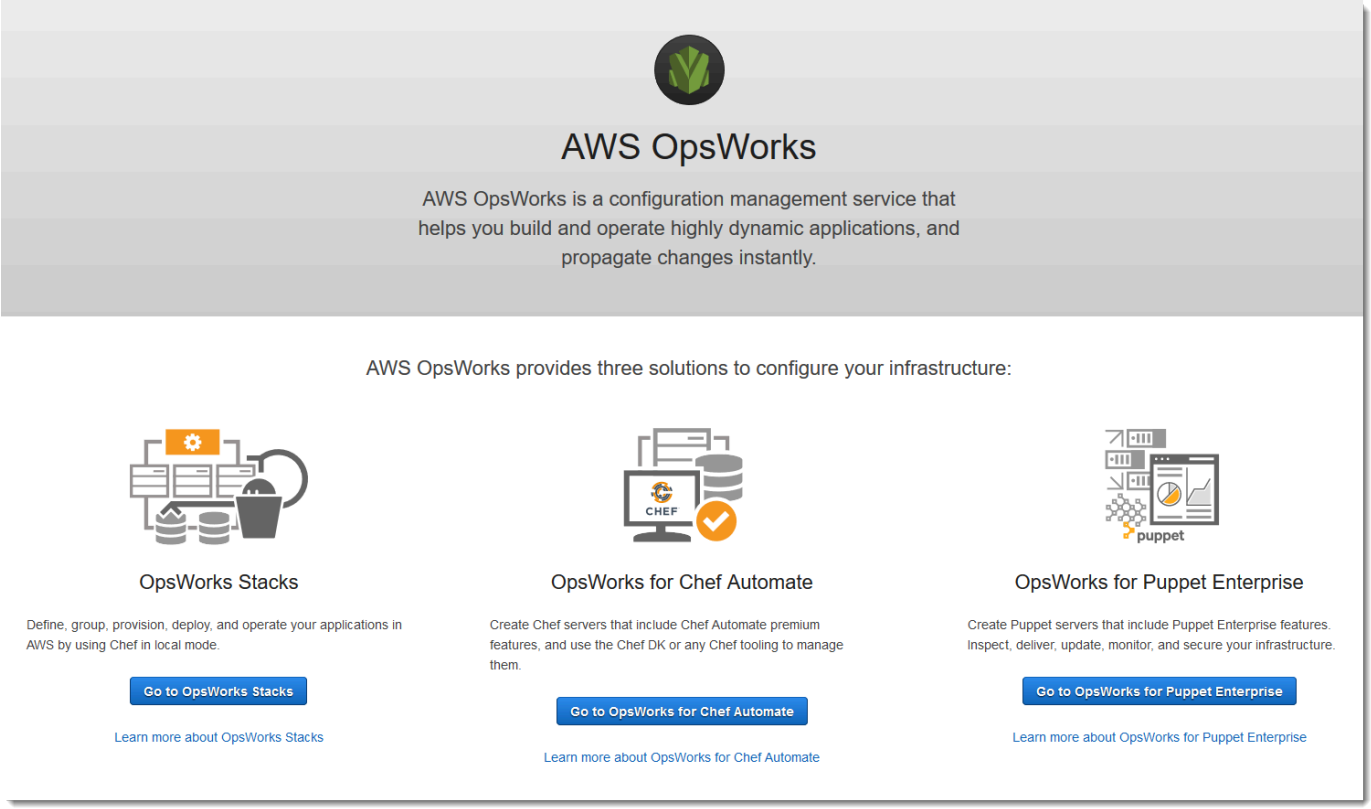
Vous pouvez créer un serveur Chef à l'aide de la AWS OpsWorks for Chef Automate console ou du AWS CLI.

Rubriques

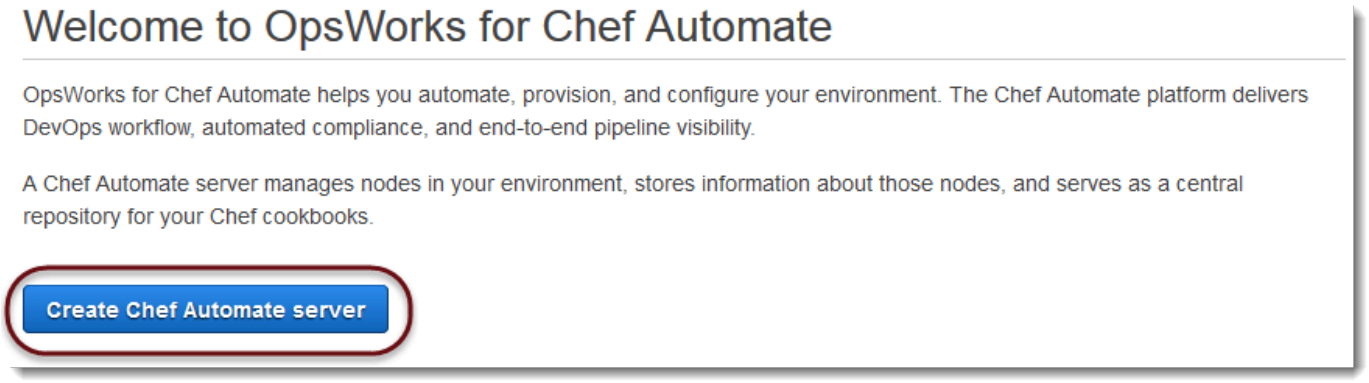
- Créez un serveur Chef Automate dans AWS Management Console
- Créez un serveur Chef Automate à l'aide du AWS CLI

Créez un serveur Chef Automate dans AWS Management Console

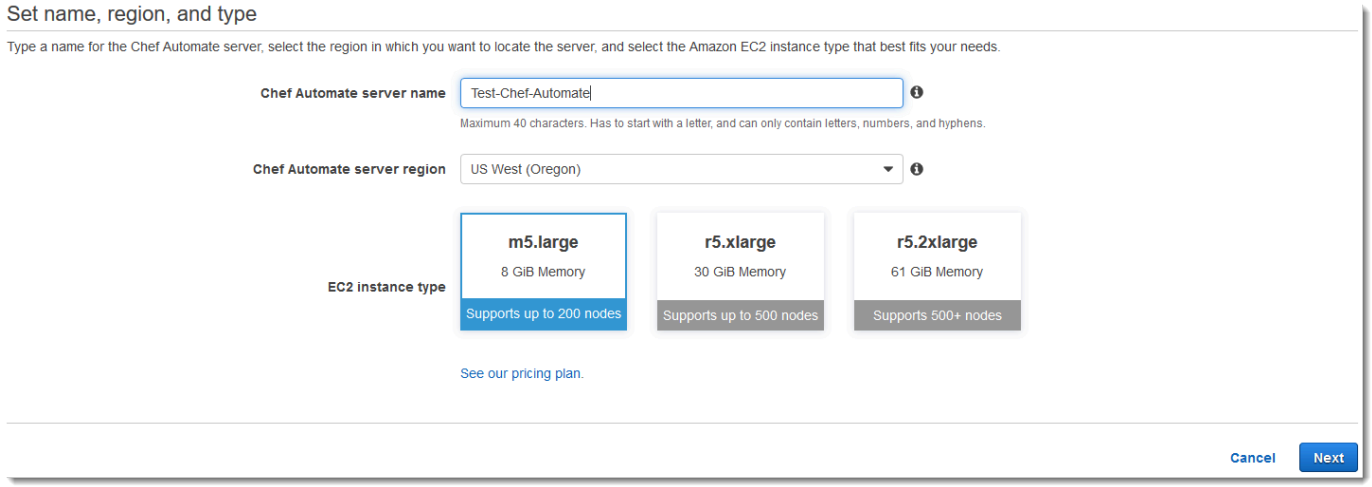
1. Connectez-vous à la AWS OpsWorks console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/opsworks/.
2. Sur la page d' AWS OpsWorks accueil, choisissez Accéder à OpsWorks pour Chef Automate.



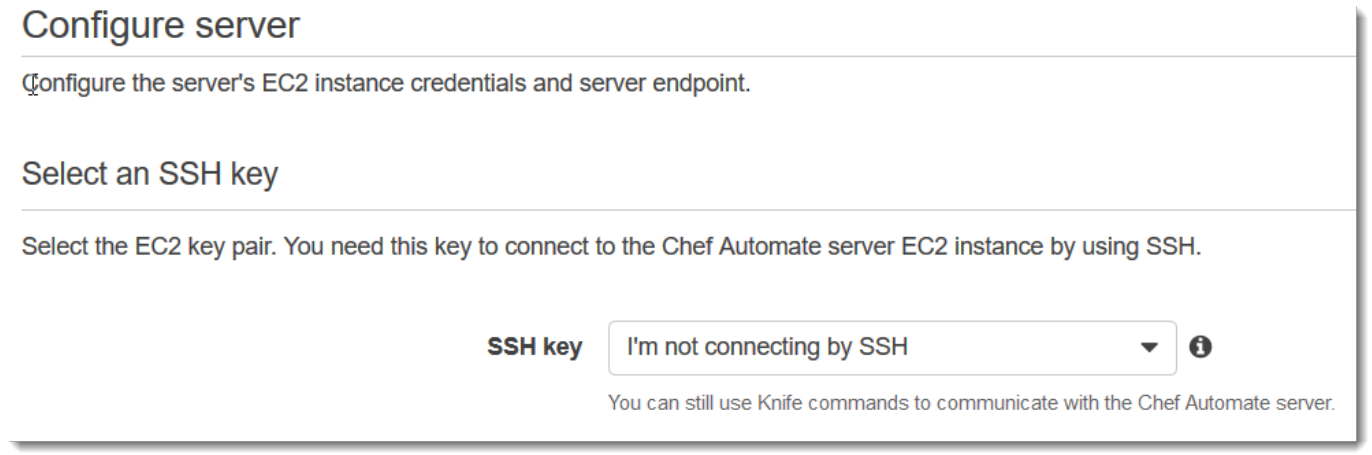
3. Sur la page d' AWS OpsWorks for Chef Automate accueil, choisissez Create Chef Automate server.



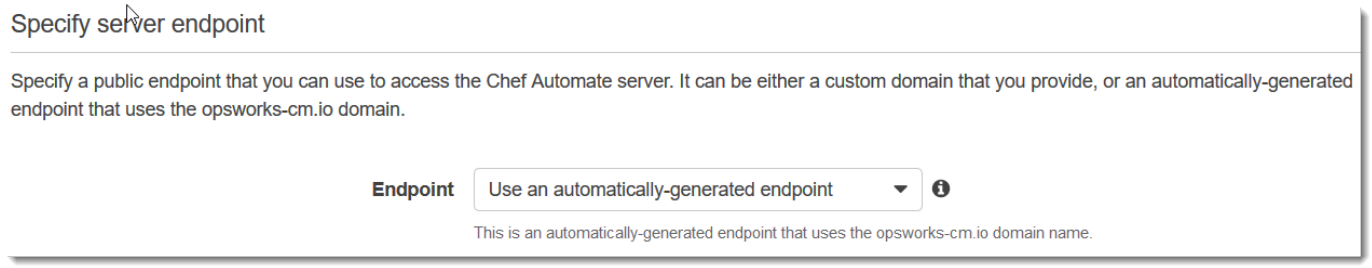
4. Sur la page Set name, region, and type (Définir le nom, la région et le type), spécifiez un nom pour votre serveur. Les noms de serveur Chef peuvent inclure 40 caractères au maximum et contenir uniquement des caractères alphanumériques et des tirets. Sélectionnez une région prise en charge, puis choisissez un type d'instance prenant en charge le nombre de nœuds que vous voulez gérer. Vous pouvez modifier le type d'instance une fois que votre serveur a été créé, si nécessaire. Pour cette procédure pas à pas, nous créons un type d'instance m5.large dans la région USA Ouest (Oregon). Choisissez Suivant.



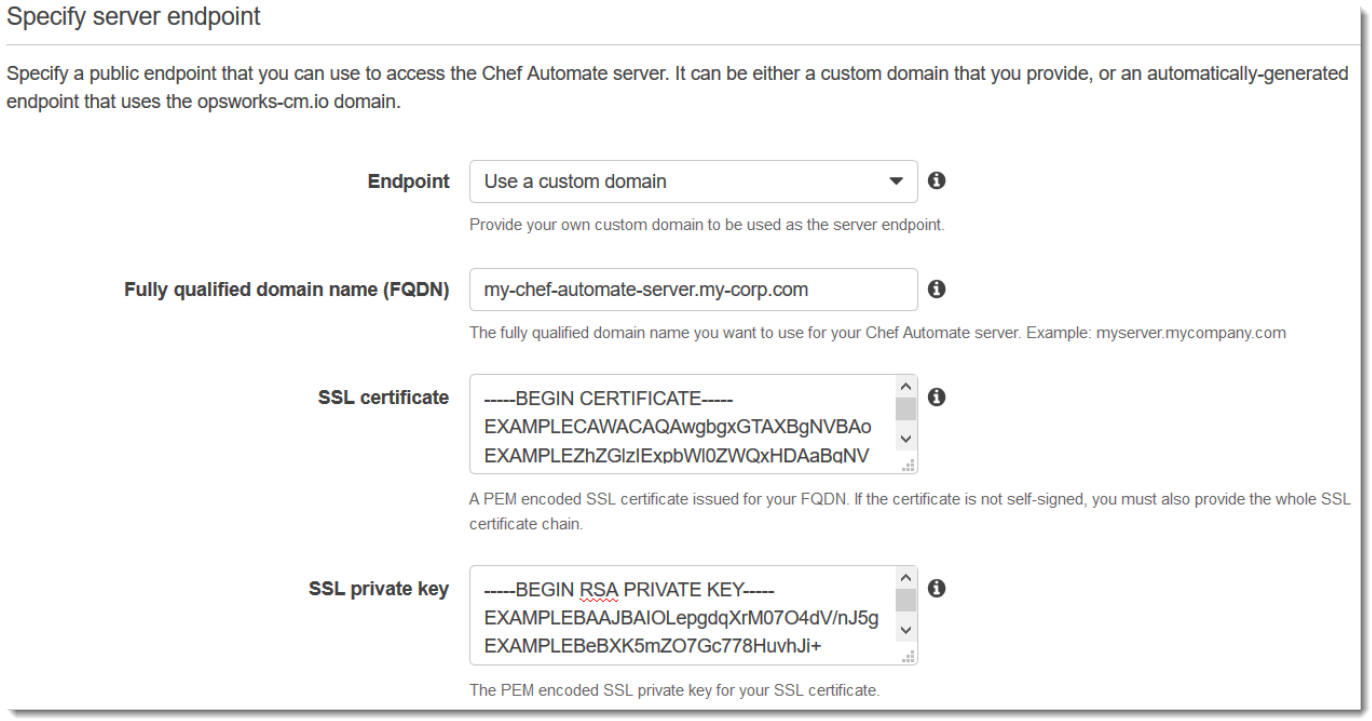
5. Sur la page Configure server (Configurer le serveur), conservez la sélection par défaut dans la liste déroulante SSH key (Clé SSH), sauf si vous souhaitez spécifier un nom de paire de clés.



6. Pour Specify server endpoint (Spécifier le point de terminaison du serveur), laissez la valeur par défaut, Use an automatically-generated endpoint (Utiliser un point de terminaison généré automatiquement), puis choisissez Next (Suivant), sauf si vous souhaitez que votre serveur se trouve sur votre propre domaine personnalisé. Pour configurer un domaine personnalisé, passez à l'étape suivante.



7. Pour utiliser un domaine personnalisé, dans Specify server endpoint (Spécifier le point de terminaison du serveur), choisissez Utiliser un domaine personnalisé (Use a custom domain) dans la liste déroulante.



   a. Pour Fully qualified domain name (FQDN) (Nom de domaine complet (FQDN)), spécifiez un nom de domaine complet. Vous devez posséder le nom de domaine que vous souhaitez utiliser.

   b. Pour SSL certificate (Certificat SSL), collez l'intégralité du certificat au format PEM, en commençant par `-----BEGIN CERTIFICATE-----` et se terminant par `-----END CERTIFICATE-----`. L'objet du certificat SSL doit correspondre au nom de domaine complet que vous avez entré à l'étape précédente.

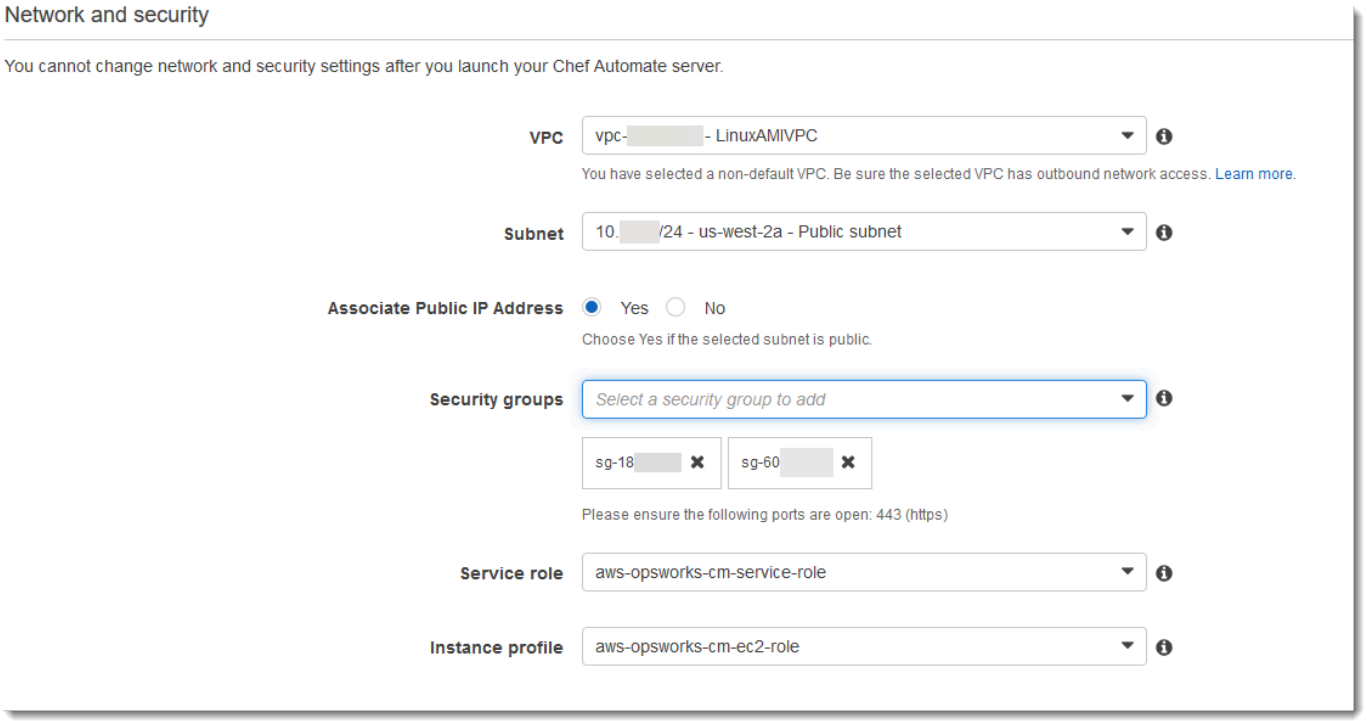
   c. Pour SSL private key (Clé privée SSL), collez l'intégralité de la clé privée RSA, en commençant par `-----BEGIN RSA PRIVATE KEY-----` et terminant par `-----END RSA PRIVATE KEY-----`. La clé privée SSL doit correspondre à la clé publique du certificat SSL que vous avez entré à l'étape précédente. Choisissez Suivant.

8. Sur la page Configuration de paramètres avancés, dans la zone Réseau et sécurité, choisissez un VPC, un sous-réseau et un ou plusieurs groupes de sécurité. Les exigences pour votre VPC sont les suivantes :

   - Le VPC doit avoir au moins un sous-réseau public.
   - La résolution DNS doit être activée.

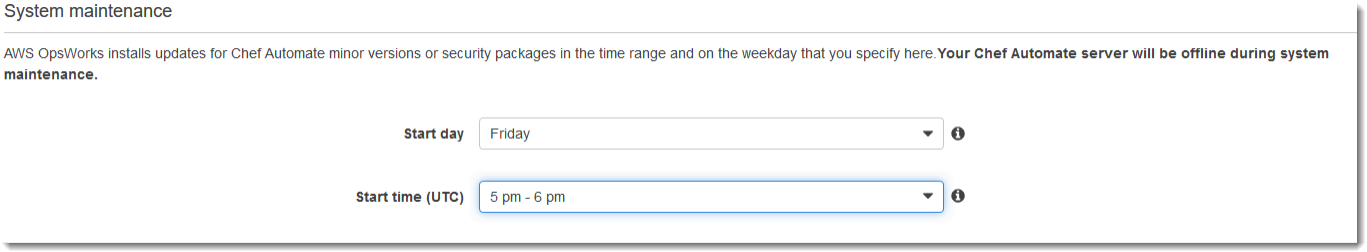
- Attribuer automatiquement l'adresse IP publique (Auto-assign public) doit être activé sur les sous-réseaux publics.

AWS OpsWorks peut générer un groupe de sécurité, un rôle de service et un profil d'instance pour vous, si vous n'en avez pas déjà un à utiliser. Votre serveur peut être membre de plusieurs groupes de sécurité. Vous ne pouvez pas modifier les paramètres de réseau et de sécurité pour le serveur Chef une fois que vous avez quitté cette page.

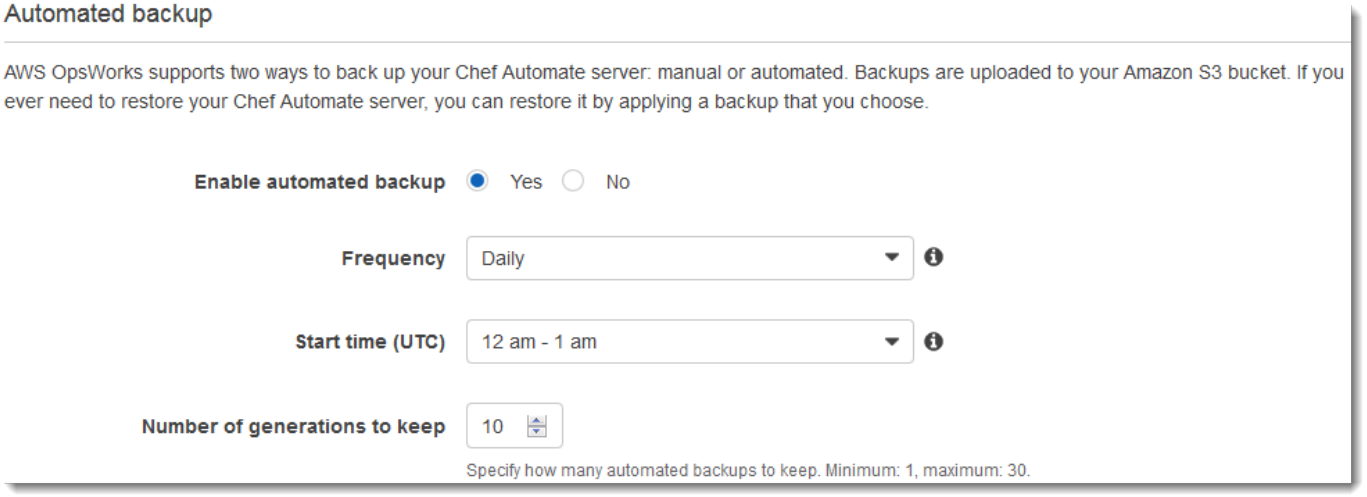


9. Dans la section System maintenance (Maintenance du système), définissez le jour et l'heure de début de la maintenance du système. Etant donné que le serveur doit être hors ligne lors de la maintenance du système, choisissez une heure où le serveur est peu sollicité pendant les heures normales de bureau. Les nœuds connectés passent à l'état `pending-server` tant que la maintenance n'est pas terminée.

   La fenêtre de maintenance est obligatoire. Vous pouvez modifier le jour et l'heure de début ultérieurement en utilisant le AWS Management Console AWS CLI, ou le APIs.



10. Configurez des sauvegardes. Par défaut, les sauvegardes automatiques sont activées. Définissez une fréquence et une heure préférées pour le démarrage de la sauvegarde automatique, et définissez le nombre de générations de sauvegarde à stocker dans Amazon Simple Storage Service. Un maximum de 30 sauvegardes sont conservées ; lorsque le maximum est atteint, les sauvegardes les plus anciennes sont AWS OpsWorks for Chef Automate supprimées pour faire de la place aux nouvelles.



11. (Facultatif) Dans Tags, ajoutez des balises au serveur et aux ressources associées, telles que l' EC2 instance, l'adresse IP élastique, le groupe de sécurité, le compartiment S3 et les sauvegardes. Pour plus d'informations sur le balisage d'un AWS OpsWorks for Chef Automate serveur, consultezUtilisation des balises sur les AWS OpsWorks for Chef Automate ressources.
12. Lorsque vous avez terminé de configurer les paramètres avancés, cliquez sur Suivant.
13. Sur la page Vérification, vérifiez vos choix. Lorsque vous êtes prêt à créer le serveur, choisissez Lancer.

    Pendant que vous attendez de AWS OpsWorks créer votre serveur Chef, accédez au kit de démarrage Configuration d'un serveur Chef à l'aide du Starter Kit et aux informations d'identification du tableau de bord Chef Automate et téléchargez-les. N'attendez pas que votre serveur soit en ligne pour télécharger ces éléments.

Lorsque la création du serveur est terminée, votre serveur Chef est disponible sur la page d'accueil d' AWS OpsWorks for Chef Automate , avec un statut online (en ligne). Une fois que le serveur est en ligne, le tableau de bord Chef Automate est disponible sur le domaine du serveur, à une URL au format suivant : `https://your_server_name-random.region.opsworks-cm.io`.

Créez un serveur Chef Automate à l'aide du AWS CLI

La création d'un AWS OpsWorks for Chef Automate serveur en exécutant AWS CLI des commandes est différente de la création d'un serveur dans la console. Dans la console, AWS OpsWorks crée un rôle de service et un groupe de sécurité pour vous, si vous ne spécifiez pas ceux que vous souhaitez utiliser. Dans le AWS CLI, vous AWS OpsWorks pouvez créer un groupe de sécurité pour vous si vous n'en spécifiez pas un, mais cela ne crée pas automatiquement un rôle de service ; vous devez fournir un ARN de rôle de service dans le cadre de votre `create-server` commande. Dans la console, lors AWS OpsWorks de la création de votre serveur Chef Automate, vous téléchargez le kit de démarrage Chef Automate et les informations de connexion pour le tableau de bord Chef Automate. Comme vous ne pouvez pas le faire lorsque vous créez un AWS OpsWorks for Chef Automate serveur à l'aide du AWS CLI, vous utilisez un utilitaire de traitement JSON pour obtenir les informations d'identification et le kit de démarrage à partir des résultats de la `create-server` commande une fois que votre nouveau AWS OpsWorks for Chef Automate serveur est en ligne. Sinon, vous pouvez générer un nouvel ensemble d'informations d'identification de connexion et un nouveau kit de démarrage dans la console une fois que votre nouveau serveur AWS OpsWorks for Chef Automate est en ligne.

Si votre ordinateur local n'exécute pas déjà le AWS CLI, téléchargez-le et installez-le AWS CLI en suivant les instructions d'installation du guide de l'utilisateur de l'interface de ligne de commande AWS. Cette section ne décrit pas tous les paramètres que vous pouvez utiliser avec la commande `create-server`. Pour plus d'informations sur les paramètres `create-server`, consultez `create-server` dans la Référence de l'AWS CLI .

1. Veillez à remplir les conditions préalables, en particulier Configuration d'un VPC, ou assurez-vous que vous disposez d'un VPC existant que vous pouvez utiliser. Pour créer votre serveur Chef Automate, vous avez besoin d'un ID de sous-réseau.
2. Le cas échéant, générez une clé centrale Chef à l'aide d'OpenSSL et enregistrez la clé dans un fichier pratique et sécurisé sur votre ordinateur local. La clé centrale est automatiquement générée dans le cadre du processus de création de serveur si vous n'en fournissez pas une

dans la commande `create-server`. Si vous souhaitez ignorer cette étape, vous pouvez obtenir la clé centrale Chef Automate à partir des résultats de la commande `create-server`. Si vous choisissez de générer la clé centrale à l'aide des commandes suivantes, veillez à inclure le paramètre `-pubout`, car la valeur de la clé centrale Chef Automate est la moitié publique de la paire de clés RSA. Pour de plus amples informations, veuillez consulter l'étape 6.

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

3. Créez un rôle de service et un profil d'instance.
4. Recherchez et copiez les rôles ARNs de service dans votre compte.

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

Dans les résultats de la commande `list-roles`, recherchez les entrées d'ARN de rôle de service ressemblant à ce qui suit. Notez le rôle du service ARNs. Ces valeurs vous seront utiles pour créer votre serveur Chef Automate.

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

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

5. Recherchez et copiez les profils ARNs d'instance de votre compte.

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

Dans les résultats de la commande `list-instance-profiles`, recherchez les entrées d'ARN de profil d'instance ressemblant à ce qui suit. Prenez note du profil de l'instance ARNs. Ces valeurs vous seront utiles pour créer votre serveur Chef Automate.

```
{
    "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. Créez le AWS OpsWorks for Chef Automate serveur en exécutant la `create-server` commande.

   - La valeur de `--engine` est `ChefAutomate`, `--engine-model` a pour valeur `Single` et `--engine-version` a pour valeur `12`.
   - Le nom du serveur doit être unique dans votre AWS compte, dans chaque région. Les noms de serveur doivent commencer par une lettre. Ensuite, vous pouvez utiliser des lettres, des chiffres ou des tirets (-), jusqu'à un maximum de 40 caractères.
   - Utilisez l'ARN du profil d'instance et celui du rôle de service que vous avez copiés lors des Étapes 4 et 5.
   - Les types d'instance valides sont `m5.large`, `r5.xlarge` et `r5.2xlarge`. Pour plus d'informations sur les spécifications de ces types d'instances, consultez la section Types d'instances du guide de EC2 l'utilisateur Amazon.
   - Le paramètre `--engine-attributes` est facultatif ; si vous ne spécifiez pas l'une des valeurs ou les deux, le processus de création de serveur les génère pour vous. Si vous ajoutez `--engine-attributes`, spécifiez la valeur `CHEF_AUTOMATE_PIVOTAL_KEY` que vous avez générée à l'étape 2, une valeur `CHEF_AUTOMATE_ADMIN_PASSWORD`, ou les deux.

     Si vous ne définissez pas de valeur pour `CHEF_AUTOMATE_ADMIN_PASSWORD`, un mot de passe est généré et inclus dans la réponse `create-server`. Vous pouvez également télécharger à nouveau le kit de démarrage dans la console, ce qui permet de générer à nouveau ce mot de passe. La longueur du mot de passe est de huit caractères au minimum et de 32 au maximum. Le mot de passe peut contenir des lettres, des nombres et des caractères spéciaux (`!/@#$%^+=_`). Il doit contenir au moins une lettre en minuscule, une lettre en majuscule, un chiffre et un caractère spécial.

- Une paire de clés SSH est facultative, mais elle peut vous aider à vous connecter à votre serveur Chef Automate si vous devez réinitialiser le mot de passe de l'administrateur du tableau de bord Chef Automate. Pour plus d'informations sur la création d'une paire de clés SSH, consultez Amazon EC2 Key Pairs dans le guide de l' EC2 utilisateur Amazon.
- Pour utiliser un domaine personnalisé, ajoutez les paramètres suivants à votre commande. Sinon, le processus de création de serveur Chef Automate génère automatiquement un point de terminaison pour vous. Les trois paramètres sont requis pour configurer un domaine personnalisé. Pour plus d'informations sur les exigences supplémentaires relatives à l'utilisation de ces paramètres, consultez CreateServerla référence de l'API AWS OpsWorks CM.
  - `--custom-domain` - Point de terminaison public facultatif d'un serveur, tel que `https://aws.my-company.com`.
  - `--custom-certificate` - Un certificat HTTPS au format PEM. La valeur peut être un seul certificat auto-signé ou une chaîne de certificats.
  - `--custom-private-key` - Une clé privée au format PEM pour se connecter au serveur à l'aide de HTTPS. La clé privée ne doit pas être chiffrée ; elle ne peut pas être protégée par un mot de passe ou une phrase passe.
- Une maintenance hebdomadaire du système est obligatoire. Vous devez spécifier des valeurs valides au format suivant : `DDD:HH:MM`. L'heure doit être exprimée en heure UTC (temps universel coordonné). Si vous ne spécifiez aucune valeur pour `--preferred-maintenance-window`, la valeur par défaut est un intervalle aléatoire d'une heure le mardi, le mercredi ou le vendredi.
- Les valeurs valides pour `--preferred-backup-window` doivent être spécifiées dans l'un des formats suivants : `HH:MM` pour les sauvegardes quotidiennes ou `DDD:HH:MM` pour les sauvegardes hebdomadaires. L'heure indiquée est en heure UTC. La valeur par défaut est une heure de début quotidienne aléatoire. Pour annuler les sauvegardes quotidiennes, ajoutez le paramètre `--disable-automated-backup`.
- Pour`--security-group-ids`, entrez un ou plusieurs groupes de sécurité IDs, séparés par un espace.
- Pour `--subnet-ids`, entrez un ID de sous-réseau.

```
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
```

Voici un exemple.

```
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
```

L'exemple suivant crée un serveur Chef Automate qui utilise un domaine personnalisé.

```
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
```

L'exemple suivant crée un serveur Chef Automate qui ajoute deux balises : `Stage: Production` et `Department: Marketing`. Pour plus d'informations sur l'ajout et la gestion de balises sur AWS OpsWorks for Chef Automate les serveurs, consultez Utilisation des balises sur les AWS OpsWorks for Chef Automate ressources ce guide.

```
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 prend environ 15 minutes pour créer un nouveau serveur. Ne supprimez pas la sortie de la commande `create-server` ou ne fermez pas votre session shell, car la sortie peut contenir des informations importantes qui ne seront plus réaffichées. Pour obtenir les mots de passe et le kit de démarrage à partir des résultats de la commande `create-server`, passez à l'étape suivante.

   Si vous utilisez un domaine personnalisé avec le serveur, dans la sortie de la commande `create-server`, copiez la valeur de l'attribut `Endpoint`. Voici un exemple.

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

8. Si vous avez choisi de AWS OpsWorks for Chef Automate générer une clé et un mot de passe pour vous, vous pouvez les extraire dans des formats utilisables à partir des `create-server` résultats à l'aide d'un processeur JSON tel que jq. Une fois que vous avez installé jq, vous pouvez exécuter les commandes suivantes pour extraire la clé centrale, le mot de passe administrateur du tableau de bord Chef Automate et le kit de démarrage. Si vous n'avez pas

fourni votre propre clé centrale et votre mot de passe à l'étape 4, veillez à sauvegarder la clé centrale et le mot de passe administrateur extraits dans des emplacements sécurisés, mais pratiques.

```
#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
```

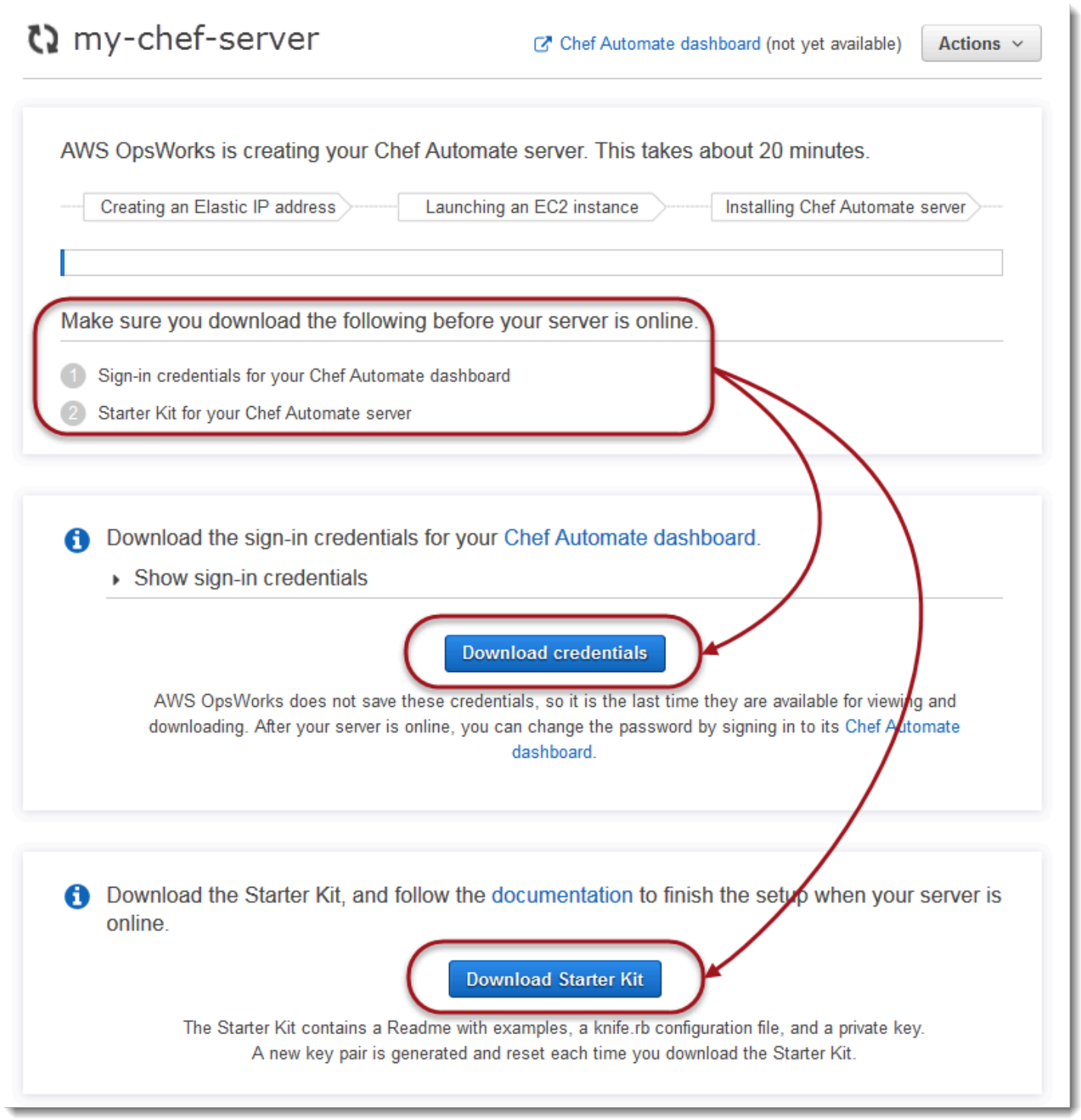
9. Si vous n'avez pas extrait le kit de démarrage à partir des résultats des `create-server` commandes, vous pouvez éventuellement télécharger un nouveau kit de démarrage depuis la page Propriétés du serveur dans la AWS OpsWorks for Chef Automate console. Le téléchargement d'un nouveau kit de démarrage réinitialise le mot de passe administrateur du tableau de bord Chef Automate.
10. Si vous n'utilisez pas un domaine personnalisé, passez à l'étape suivante. Si vous utilisez un domaine personnalisé avec le serveur, créez une entrée CNAME dans l'outil de gestion DNS de votre entreprise pour faire pointer votre domaine personnalisé vers le point de AWS OpsWorks for Chef Automate terminaison que vous avez copié à l'étape 7. Vous ne pouvez pas accéder ou vous connecter à un serveur avec un domaine personnalisé tant que vous n'avez pas effectué cette étape.
11. Lorsque le processus de création du serveur est terminé, accédez à the section called “Fin de la configuration et chargement des livres de recettes”.

Configuration d'un serveur Chef à l'aide du Starter Kit

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants

de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Alors que la création du serveur Chef est encore en cours, ouvrez sa page de propriétés dans la console AWS OpsWorks for Chef Automate . La première fois que vous utilisez un nouveau serveur Chef, la page de propriétés vous invite à télécharger deux éléments requis. Téléchargez ces éléments avant que votre serveur Chef soit en ligne ; les boutons de téléchargement ne sont pas disponibles une fois qu'un nouveau serveur est en ligne.



- Informations d'identification de connexion pour le serveur Chef. Vous utiliserez ces informations d'identification pour vous connecter au tableau de bord Chef Automate, où vous utiliserez les fonctionnalités premium de Chef Automate, telles que les analyses de flux de travail et de conformité. AWS OpsWorks n'enregistre pas ces informations d'identification ; c'est la dernière fois

qu'elles sont disponibles pour consultation et téléchargement. Si nécessaire, vous pouvez modifier le mot de passe qui est fourni avec ces informations d'identification une fois connecté.

- Starter Kit. Le Starter Kit contient un fichier README avec des exemples, un fichier de configuration `knife.rb` et une clé privée pour l'utilisateur principal ou central. Une nouvelle paire de clés est générée, et l'ancienne clé est réinitialisée, chaque fois que vous téléchargez le Starter Kit.

Outre les informations d'identification qui ne fonctionnent qu'avec le nouveau serveur, le fichier .zip du Starter Kit inclut un exemple simple de référentiel Chef compatible avec n'importe quel AWS OpsWorks for Chef Automate serveur. Dans le référentiel Chef, vous stockez les livres de recettes, les rôles, les fichiers de configuration et autres artefacts pour la gestion de vos nœuds avec Chef. Nous vous recommandons de stocker ce référentiel dans un système de contrôle de version, comme Git, et de le traiter comme code source. Pour plus d'informations et d'exemples sur la configuration d'un référentiel Chef qui est suivi dans Git, consultez la page About the chef-repo dans la documentation Chef.

Prérequis

1. Alors que la création du serveur est encore en cours, téléchargez les informations d'identification de connexion pour le serveur Chef et sauvegardez-les dans un emplacement sécurisé, mais pratique.
2. Téléchargez le Starter Kit et décompressez le fichier .zip Starter Kit dans votre répertoire d'espace de travail. Ne partagez pas la clé privée Starter Kit. Si d'autres utilisateurs gèrent le serveur Chef, ajoutez-les comme administrateurs du tableau de bord Chef Automate plus tard.
3. Téléchargez et installez Chef Workstation (anciennement connu sous le nom de Chef Development Kit, ou Chef DK) sur l'ordinateur que vous utiliserez pour gérer votre serveur et vos nœuds Chef. L'`knife`utilitaire fait partie de Chef Workstation. Pour obtenir des instructions, voir Installer Chef Workstation sur le site Web de Chef.

Exploration du contenu du Starter Kit

Le contenu du Starter Kit est le suivant.

- `cookbooks/` - Répertoire des livres de recettes que vous créez. Le `cookbooks/` dossier contient le livre de `opsworks-webserver` recettes, un livre de recettes emballé qui dépend du livre de `nginx` recettes du site Web de Chef Supermarket. `Policyfile.rb`utilise par défaut le

supermarché Chef comme source secondaire si les dépendances des livres de recettes ne sont pas disponibles dans le `cookbooks/` répertoire.

- `Policyfile.rb` - Fichier de stratégie basé sur Ruby qui définit les livres de recettes, les dépendances et les attributs qui deviennent la stratégie de vos nœuds.
- `userdata.sh` et `userdata.ps1` - Vous pouvez utiliser des fichiers de données d'utilisateur pour associer automatiquement des nœuds après le lancement de votre serveur Chef Automate. `userdata.sh` est destiné à l'amorçage des nœuds basés sur Linux, et `userdata.ps1` est destiné aux nœuds basés sur Windows.
- `Berksfile` - Vous pouvez utiliser ce fichier si vous préférez utiliser Berkshelf et des commandes `berks` pour charger les livres de recettes et leurs dépendances. Dans cette procédure, nous utilisons `Policyfile.rb` et des commandes Chef pour charger les livres de recettes, les dépendances et les attributs.
- `README.md` - Fichier basé sur Markdown qui explique comment utiliser le Starter Kit pour configurer votre serveur Chef Automate pour la première fois.
- `.chef` est un répertoire masqué qui contient un fichier de configuration knife (`knife.rb`) et un fichier de clé d'authentification secrète (.pem).
  - `.chef/knife.rb` - Fichier de configuration knife (`knife.rb`). Le `knife.rb`fichier est configuré de telle sorte que les opérations de l'`knife`outil Chef s'exécutent sur le AWS OpsWorks for Chef Automate serveur.
  - `.chef/ca_certs/opsworks-cm-ca-2020-root.pem` - Clé privée SSL signée par une autorité de certification (CA) fournie par AWS OpsWorks. Cette clé autorise le serveur à s'identifier auprès de l'agent du client Chef Infra sur les nœuds gérés par votre serveur.

Configuration de votre référentiel Chef

Un référentiel Chef contient plusieurs répertoires. Chaque répertoire du Starter Kit contient un fichier README qui décrit l'objectif du répertoire et comment utiliser celui-ci pour gérer vos systèmes avec Chef. Il existe deux façons d'installer des livres de recettes sur votre serveur Chef : en exécutant des commandes `knife` ou en exécutant une commande Chef pour charger sur votre serveur un fichier de stratégie (`Policyfile.rb`) qui télécharge et installe les livres de recettes spécifiés. Cette procédure utilise des commandes Chef et un fichier `Policyfile.rb` pour installer des livres de recettes sur votre serveur.

1. Créez un répertoire sur votre ordinateur local pour stocker les livres de recettes, par exemple `chef-repo`. Après avoir ajouté des livres de recettes, des rôles et d'autres fichiers à ce

référentiel, nous vous recommandons de le télécharger ou de le stocker dans un système sécurisé et versionné, tel que CodeCommit Git ou Amazon S3.

2. Dans le répertoire `chef-repo`, créez les répertoires suivants :

   - `cookbooks/`- Permet de ranger des livres de cuisine.
   - `roles/` - Stocke les rôles au format `.rb` ou `.json`.
   - `environments/` - Stocke les environnements au format `.rb` ou `.json`.

Utilisation du fichier Policyfile.rb pour obtenir des livres de recettes d'une source à distance

Dans cette section, vous allez modifier `Policyfile.rb` pour spécifier des livres de recettes, puis exécuter une commande Chef afin de charger le fichier sur votre serveur et d'installer les livres de recettes.

1. Affichez le fichier `Policyfile.rb` de votre Starter Kit. Par défaut, `Policyfile.rb` inclut le livre de recettes wrapper `opsworks-webserver`, qui dépend du livre de recettes `nginx` disponible sur le site web Chef Supermarket. Le livre de recettes `nginx` installe et configure un serveur web sur les nœuds gérés. Le livre de recettes `chef-client` requis, qui installe l'agent du client Chef Infra sur les nœuds gérés, est également spécifié.

   `Policyfile.rb` pointe également sur le livre de recettes Chef Audit facultatif, que vous pouvez utiliser pour configurer des analyses de conformité sur les nœuds. Pour plus d'informations sur la configuration des analyses de conformité et l'obtention de résultats de conformité pour les nœuds gérés, consultez Analyses de conformité dans AWS OpsWorks for Chef Automate. Si vous ne souhaitez pas configurer les fonctions d'analyse de conformité et d'audit maintenant, supprimez `'audit'` de la section `run_list` et ne spécifiez pas les attributs de livre de recettes `audit` à la fin du fichier.

```
# 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"
  }
]
```

Voici un exemple de fichier `Policyfile.rb` sans le livre de recettes et les attributs `audit`, si vous souhaitez configurer uniquement le serveur web `nginx` pour l'instant.

```
# 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'
```

Si vous apportez des modifications à `Policyfile.rb`, veillez à enregistrer le fichier.

2. Téléchargez et installez les livres de recettes définis dans `Policyfile.rb`.

```
chef install
```

Tous les livres de recettes sont versionnés dans le fichier `metadata.rb` du livre de recettes. Chaque fois que vous modifiez un livre de recettes, vous devez modifier la version du livre de recettes se trouvant dans son fichier `metadata.rb`.

3. Si vous avez choisi de configurer les analyses de conformité, et conservé les informations sur le livre de recettes `audit` dans le fichier de stratégie, transmettez la stratégie `opsworks-demo` à votre serveur.

```
chef push opsworks-demo
```

4. Si vous avez terminé l'étape 3, vérifiez l'installation de votre stratégie. Exécutez la commande suivante.

```
chef show-policy
```

Les résultats doivent ressembler à ce qui suit :

```
opsworks-demo-webserver
=======================
```

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

5. Vous êtes maintenant prêt à ajouter ou amorcer des nœuds sur votre serveur Chef Automate. Vous pouvez automatiser l'association des nœuds en suivant les étapes décrites dans Ajouter des nœuds automatiquement AWS OpsWorks for Chef Automate ou ajouter des nœuds l'un après l'autre en suivant les étapes indiquées dans Ajouter des nœuds individuellement.

(Alternative) Utilisation de Berkshelf pour obtenir des livres de recettes d'une source à distance

Berkshelf est un outil de gestion de livres de recettes et de leurs dépendances. Si vous préférez utiliser Berkshelf au lieu de `Policyfile.rb` pour installer des livres de recettes dans un stockage local, utilisez la procédure décrite dans cette section au lieu de la section précédente. Vous pouvez spécifier les livres de recettes et les versions à utiliser avec votre serveur Chef et les charger. Le Starter Kit contient un fichier nommé `Berksfile` que vous pouvez utiliser pour répertorier vos livres de recettes.

1. Pour commencer, ajoutez le livre de recettes `chef-client` au fichier Berksfile qui est inclus. Le livre de recettes `chef-client` configure le logiciel agent du client Chef Infra sur chaque nœud que vous connectez à votre serveur Chef Automate. Pour en savoir plus sur ce livre de recettes, consultez Livre de recettes du client Chef sur le site Chef Supermarket.
2. À l'aide d'un éditeur de texte, ajoutez à votre fichier Berksfile un autre livre de recettes qui installe une application de serveur web, par exemple, le livre de recettes `apache2`, qui installe le serveur web Apache. Votre fichier Berksfile doit ressembler à ce qui suit.

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

3. Téléchargez et installez les livres de recettes sur votre ordinateur local.

   ```
   berks install
   ```

4. Chargez le livre de recettes sur le serveur Chef.

   Sous Linux, exécutez la commande suivante.

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

Sous Windows, exécutez la commande Chef Workstation suivante dans une PowerShell session. Avant d'exécuter la commande, veillez à définir la politique PowerShell d'exécution sur`RemoteSigned`. Ajoutez `chef shell-init` pour rendre les commandes de l'utilitaire Chef Workstation accessibles à 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. Vérifiez l'installation du livre de recettes en affichant une liste de livres de recettes actuellement disponibles sur le serveur Chef Automate. Pour ce faire, exécutez la commande `knife` suivante :

   Vous êtes prêt à ajouter des nœuds à gérer avec le AWS OpsWorks for Chef Automate serveur.

   ```
   knife cookbook list
   ```

(Facultatif) Configurer **`knife`** pour utiliser un domaine personnalisé

Si votre serveur Chef Automate utilise un domaine personnalisé, vous devrez peut-être ajouter le certificat PEM de l'autorité de certification racine qui a signé la chaîne de certificats de votre serveur, ou le certificat PEM de votre serveur si le certificat est auto-signé. `ca_certs`est un sous-répertoire `chef/` qui contient les autorités de certification (CAs) approuvées par l'`knife`utilitaire Chef.

Vous pouvez ignorer cette section si vous n'utilisez pas un domaine personnalisé ou si votre certificat personnalisé est signé par une CA racine approuvée par votre système d'exploitation. Sinon, configurez `knife` pour approuver le certificat SSL de votre serveur Chef Automate, comme décrit dans les étapes suivantes.

1. Exécutez la commande suivante.

   ```
   knife ssl check
   ```

   Si les résultats sont similaires aux suivants, ignorez le reste de cette procédure et passez à Ajout de nœuds à gérer avec le serveur Chef.

   ```
   Connecting to host my-chef-automate-server.my-corp.com:443
   ```

```
          Successfully verified certificates from 'my-chef-automate-server.my-
corp.com'
```

Si vous recevez un message d'erreur similaire au suivant, passez à l'étape suivante.

```
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. Exécutez `knife ssl fetch` pour approuver les certificats de votre serveur AWS OpsWorks for Chef Automate . Vous pouvez également copier manuellement le certificat au format PEM de la CA racine de votre serveur dans le répertoire qui est la valeur de `trusted_certs_dir` dans la sortie de `knife ssl check`. Par défaut, ce répertoire se trouve `.chef/ca_certs/` dans le Starter Kit. Votre sortie doit ressembler à ce qui suit :

```
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. Exécutez à nouveau `knife ssl check`. Votre sortie doit ressembler à ce qui suit :

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

Vous êtes prêt à utiliser `knife` avec votre serveur Chef Automate.

Ajout de nœuds à gérer avec le serveur Chef

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

L'agent `chef-client` exécute des recettes Chef sur les ordinateurs physiques ou virtuels, appelés nœuds, qui sont associés au serveur. Vous pouvez connecter des ordinateurs ou des instances sur site au serveur Chef à gérer, à condition que les nœuds exécutent des systèmes d'exploitation pris en charge. L'enregistrement des nœuds auprès du serveur Chef installe le logiciel agent `chef-client` sur ces nœuds.

Vous pouvez utiliser les méthodes suivantes pour ajouter des nœuds :

- Ajoutez des notes individuellement en exécutant une `knife` commande qui ajoute, ou amorce, une EC2 instance afin que le serveur Chef puisse la gérer. Pour de plus amples informations, veuillez consulter Ajouter des nœuds individuellement.
- Ajoutez des nœuds automatiquement à l'aide d'un script pour effectuer une association sans surveillance des nœuds avec le serveur Chef. Le code du Starter Kit montre comment ajouter automatiquement des nœuds à l'aide de la méthode sans surveillance. Pour plus d'informations, voir,Ajouter des nœuds automatiquement AWS OpsWorks for Chef Automate.

Rubriques

- Ajouter des nœuds individuellement
- Ajouter des nœuds automatiquement AWS OpsWorks for Chef Automate

Ajouter des nœuds individuellement

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants

de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section décrit comment exécuter une `knife` commande qui ajoute, ou amorce, une EC2 instance afin que le serveur Chef puisse la gérer.

La version minimale de `chef-client` prise en charge sur les nœuds associés à un serveur AWS OpsWorks for Chef Automate est 13.x. Nous vous recommandons d'utiliser la `chef-client` version stable la plus récente.

Rubriques

- (Facultatif) Spécifier l'URL de la CA racine de votre serveur Chef Automate
- Systèmes d'exploitation pris en charge
- Ajout de nœuds avec Knife

(Facultatif) Spécifier l'URL de la CA racine de votre serveur Chef Automate

Si votre serveur utilise un domaine et un certificat personnalisés, vous devrez peut-être modifier la variable `ROOT_CA_URL` dans le script userdata avec une URL publique que vous pouvez utiliser pour obtenir le certificat au format PEM de la CA racine de votre serveur. Les AWS CLI commandes suivantes chargent votre autorité de certification racine dans un compartiment Amazon S3 et génèrent une URL présignée que vous pouvez utiliser pendant une heure.

1. Téléchargez le certificat au format PEM de la CA racine dans S3.

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

2. Générez une URL pré-signée que vous pouvez utiliser pendant une heure (3600 secondes, dans cet exemple) pour télécharger la CA racine.

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

3. Modifiez la variable `ROOT_CA_URL` dans le script userdata avec la valeur de l'URL pré-signée.

Systèmes d'exploitation pris en charge

Pour obtenir la liste actuelle des systèmes d'exploitation pris en charge pour les nœuds, consultez le site web Chef.

Ajout de nœuds avec Knife

Le `knife-ec2`plug-in est inclus dans Chef Workstation. Si vous le connaissez mieux`knife-ec2`, vous pouvez l'utiliser au lieu de `knife bootstrap` provisionner et de démarrer de nouvelles EC2instances applications. Sinon, lancez une nouvelle EC2 instance, puis suivez les étapes décrites dans cette section.

Pour ajouter des nœuds à gérer

1. Exécutez la commande suivante `knife bootstrap`. Cette commande démarre une EC2 instance vers les nœuds que votre serveur Chef gérera. Notez que vous donnez au serveur Chef l'instruction d'exécuter des recettes à partir du livre de recettes `nginx` que vous avez installé dans the section called "Utilisation du fichier Policyfile.rb pour obtenir des livres de recettes d'une source à distance". Pour plus d'informations sur l'ajout de nœuds en exécutant la commande `knife bootstrap`, consultez Amorçage d'un nœud dans la documentation Chef.

   Le tableau suivant présente les noms d'utilisateur valides pour les systèmes d'exploitation de nœuds de la commande `knife` au cours de cette étape. Si ni `root`, ni `ec2-user` ne fonctionne, vérifiez auprès de votre fournisseur d'AMI. Pour plus d'informations sur la connexion aux instances basées sur Linux, consultez Connexion à votre instance Linux à l'aide de SSH dans la documentation AWS.

   Valeurs valides pour les noms d'utilisateur dans les systèmes d'exploitation de nœuds

Système d'exploitation	Noms d'utilisateur valides
Amazon Linux	`ec2-user`
Red Hat Enterprise Linux 5	`root` ou `ec2-user`
Ubuntu	`ubuntu`
Fedora	`fedora` ou `ec2-user`
SUSE Linux	`root` ou `ec2-user`

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

2. Vérifiez que le nouveau nœud a été ajouté en exécutant les commandes suivantes, en le *INSTANCE_NAME* remplaçant par le nom de l'instance que vous venez d'ajouter.

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

Ajouter des nœuds automatiquement AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette rubrique explique comment ajouter automatiquement des nœuds Amazon Elastic Compute Cloud (Amazon EC2) à votre serveur Chef. Le code du Starter Kit montre comment ajouter automatiquement des nœuds à l'aide de la méthode sans surveillance. La méthode recommandée d'association sans surveillance (ou automatique) de nouveaux nœuds consiste à configurer le livre de recettes Chef Client. Vous pouvez utiliser le script `userdata` du Starter Kit et modifier la section `run_list` du script `userdata`, ou utiliser votre `Policyfile.rb` avec les livres de recettes que vous souhaitez appliquer à vos nœuds. Avant d'exécuter l'agent `chef-client`, installez le livre de recettes Chef Client sur votre serveur Chef, puis installez l'agent `chef-client` en mode de service avec, par exemple, un rôle HTTPD, comme illustré dans l'exemple de commande suivant.

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

Afin de communiquer avec le serveur Chef, le logiciel agent `chef-client` doit avoir accès à la clé publique du nœud client. Vous pouvez générer une paire de clés publique-privée dans Amazon EC2, puis transmettre la clé publique à l'appel d' AWS OpsWorks `associate-node`API avec le nom du nœud. Le script inclus dans le Starter Kit collecte automatiquement vos nom d'organisation, nom de

serveur et point de terminaison de serveur. Ceci garantit que le nœud est associé au serveur Chef et que le logiciel agent `chef-client` qui s'exécute sur ce nœud peut communiquer avec le serveur après la validation de la clé privée.

La version minimale de `chef-client` prise en charge sur les nœuds associés à un serveur AWS OpsWorks for Chef Automate est 13.x. Nous vous recommandons d'utiliser la `chef-client` version stable la plus récente.

Pour plus d'informations sur la façon de dissocier un nœud, consultez Dissocier un nœud d'un serveur AWS OpsWorks for Chef Automate ce guide et la documentation `disassociate-node`de l' AWS OpsWorks for Chef Automate API.

Rubriques

- Systèmes d'exploitation pris en charge
- Étape 1 : créer un rôle IAM à utiliser comme profil d'instance
- Étape 2 : Installation du livre de recettes Chef Client
- Étape 3 : Création d'instances en utilisant un script d'association sans surveillance
- Autres méthodes d'automatisation des exécutions répétées de chef-client
- Rubriques connexes

Systèmes d'exploitation pris en charge

Pour obtenir la liste actuelle des systèmes d'exploitation pris en charge pour les nœuds, consultez le site web Chef.

Étape 1 : créer un rôle IAM à utiliser comme profil d'instance

Créez un rôle AWS Identity and Access Management (IAM) à utiliser comme profil d' EC2 instance et associez la politique suivante au rôle IAM. Cette politique permet à l'API AWS OpsWorks for Chef Automate (`opsworks-cm`) de communiquer avec l' EC2 instance lors de l'enregistrement du nœud. Pour plus d'informations sur les profils d'instance, consultez la section Utilisation des profils d'instance dans la EC2 documentation Amazon. Pour plus d'informations sur la création d'un rôle IAM, consultez la section Création d'un rôle IAM dans la console dans la documentation Amazon EC2 .

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

```
        {
            "Action": [
                "opsworks-cm:AssociateNode",
                "opsworks-cm:DescribeNodeAssociationStatus",
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Étape 2 : Installation du livre de recettes Chef Client

Si vous ne l'avez pas déjà fait, suivez les étapes de la rubrique (Alternative) Utilisation de Berkshelf pour obtenir des livres de recettes d'une source à distance pour vous assurer que votre fichier Berksfile ou `Policyfile.rb` fasse référence au livre de recettes Chef Client et installe ce livre de recettes.

Étape 3 : Création d'instances en utilisant un script d'association sans surveillance

1. Pour créer des EC2 instances, vous pouvez copier le `userdata` script depuis le kit de démarrage vers la `userdata` section des instructions relatives aux EC2 instances, les configurations de lancement du groupe Amazon EC2 Auto Scaling ou un AWS CloudFormation modèle. Pour plus d'informations sur l'ajout de scripts aux données utilisateur, consultez Exécuter des commandes sur votre instance Linux au lancement dans la EC2 documentation Amazon.

   Ce script exécute la `associate-node`commande `opsworks-cm` API pour associer un nouveau nœud à votre serveur Chef.

   Par défaut, le nom du nouveau nœud inscrit est l'ID d'instance, mais vous pouvez changer ce nom en modifiant la valeur de la variable `NODE_NAME` dans le script `userdata`. Dans la mesure où il est actuellement impossible de changer le nom de l'organisation dans l'interface utilisateur de la console Chef, laissez `CHEF_AUTOMATE_ORGANIZATION` défini sur `default`.
2. Suivez la procédure décrite dans la section Lancement d'une instance dans la EC2 documentation, avec les modifications ici. Dans l'assistant de lancement d' EC2 instance, choisissez une AMI Amazon Linux.
3. Sur la page Configurer les détails de l'instance, sélectionnez le rôle que vous avez créé à la section Étape 1 : créer un rôle IAM à utiliser comme profil d'instance, comme étant votre rôle IAM.

4. Dans la zone Détails avancés, chargez le script `userdata.sh` que vous avez créé précédemment dans cette procédure.
5. Aucune modification n'est nécessaire sur la page Add Storage. Accédez à la page Add Tags.
6. Sur la page Configurer le groupe de sécurité, sélectionnez Ajouter une règle, puis choisissez le type HTTP pour ouvrir les ports 443 et 80 pour le serveur web Apache de cet exemple.
7. Sélectionnez Review and Launch, puis Launch. Lorsque votre nouveau nœud démarre, il applique les configurations spécifiées par les recettes que vous avez indiquées dans le paramètre `RUN_LIST`.
8. Facultatif : si vous avez ajouté le livre de recettes `nginx` à votre liste d'exécution, lorsque vous ouvrez la page web liée au DNS public de votre nouveau nœud, vous devez voir un site web qui est hébergé par votre serveur web nginx.

Autres méthodes d'automatisation des exécutions répétées de **`chef-client`**

Bien que cela soit plus difficile à réaliser et déconseillé, vous pouvez exécuter le script décrit dans cette rubrique uniquement dans le cadre des données utilisateur d'une instance autonome, utiliser un AWS CloudFormation modèle pour l'ajouter aux nouvelles données utilisateur de l'instance, configurer une `cron` tâche pour exécuter le script régulièrement ou l'exécuter `chef-client` au sein d'un service. Toutefois, nous vous recommandons la méthode du livre de recettes Chef Client car les autres techniques d'automatisation présentent certains inconvénients.

Pour obtenir la liste complète des paramètres que vous pouvez fournir à `chef-client`, consultez la documentation Chef.

Rubriques connexes

Les articles de AWS blog suivants offrent plus d'informations sur l'association automatique de nœuds à votre serveur Chef Automate, à l'aide de groupes Auto Scaling ou au sein de plusieurs comptes.

- Utilisation d'AWS OpsWorks pour Chef Automate pour gérer les EC2 instances avec Auto Scaling
- OpsWorks pour Chef Automate — Démarrage automatique des nœuds dans différents comptes

Connexion au tableau de bord Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une fois que vous avez téléchargé les informations d'identification de connexion de la page de propriétés du serveur Chef, et que le serveur est en ligne, connectez-vous au tableau de bord Chef Automate. Dans cette procédure, nous vous avons demandé de télécharger tout d'abord les livres de recettes et d'ajouter au moins un nœud à gérer. Vous pouvez ainsi consulter les informations sur les livres de recettes et sur les nœuds dans le tableau de bord.

Lorsque vous essayez de vous connecter à la page Web du tableau de bord, des avertissements relatifs aux certificats apparaissent dans votre navigateur jusqu'à ce que vous installiez un certificat SSL signé par une autorité de certification AWS OpsWorks spécifique sur l'ordinateur client que vous utilisez pour gérer votre serveur Chef. Si vous préférez ne pas voir les avertissements avant de continuer vers la page web du tableau de bord, installez le certificat SSL avant de vous connecter.

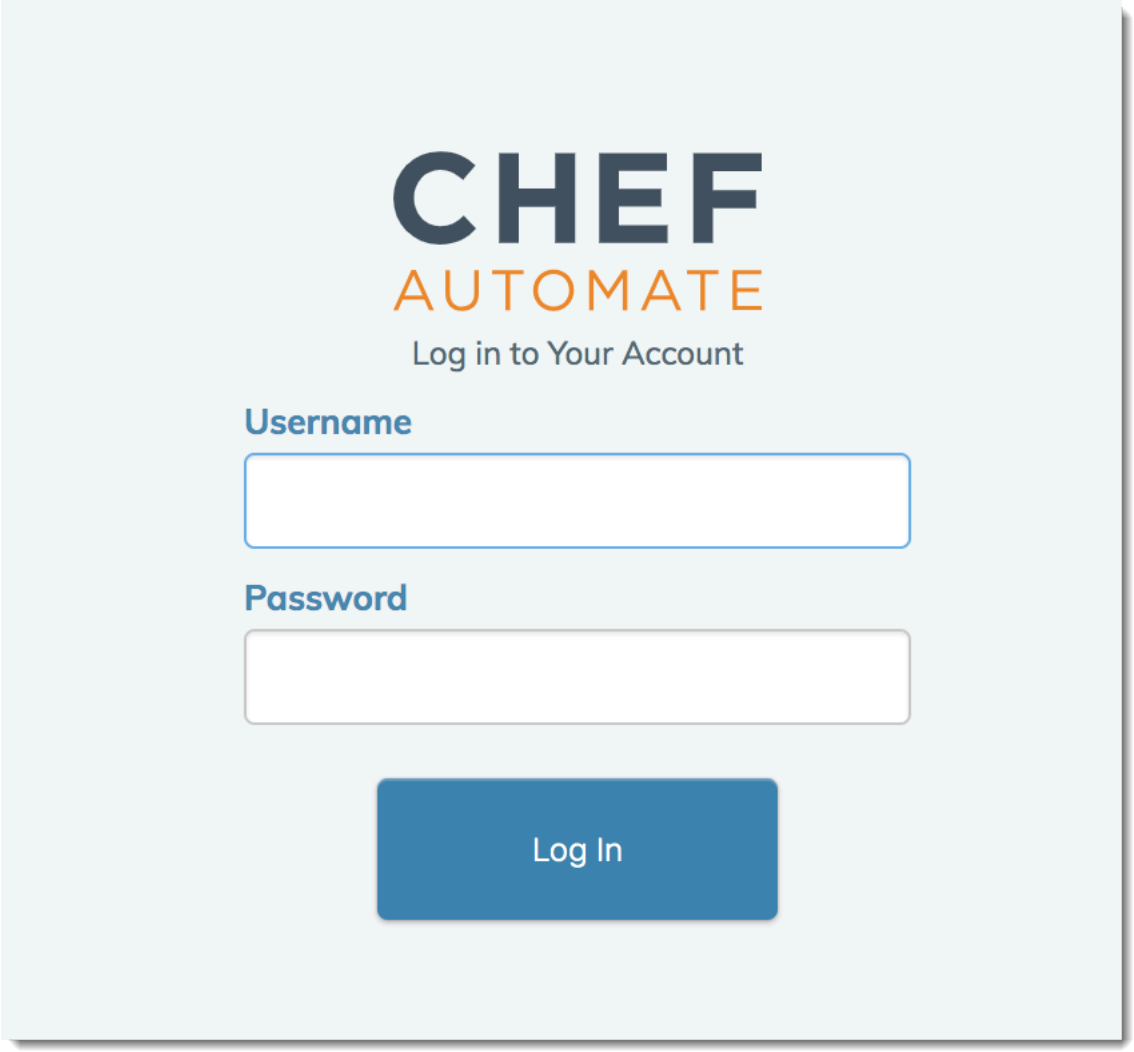
Une fois que vous avez installé le certificat SSL côté client, vous pouvez vous connecter au tableau de bord Chef Automate sans voir les messages d'avertissement.

> **Note**
>
> Les utilisateurs de Google Chrome sur les systèmes d'exploitation Ubuntu et Linux Mint peuvent avoir des difficultés à se connecter. Nous vous recommandons d'utiliser Mozilla Firefox ou d'autres navigateurs pour vous connecter et d'utiliser le tableau de bord Chef Automate sur ces systèmes d'exploitation. Aucun problème n'a été trouvé pour l'utilisation de Google Chrome sous Windows ou Mac OS.

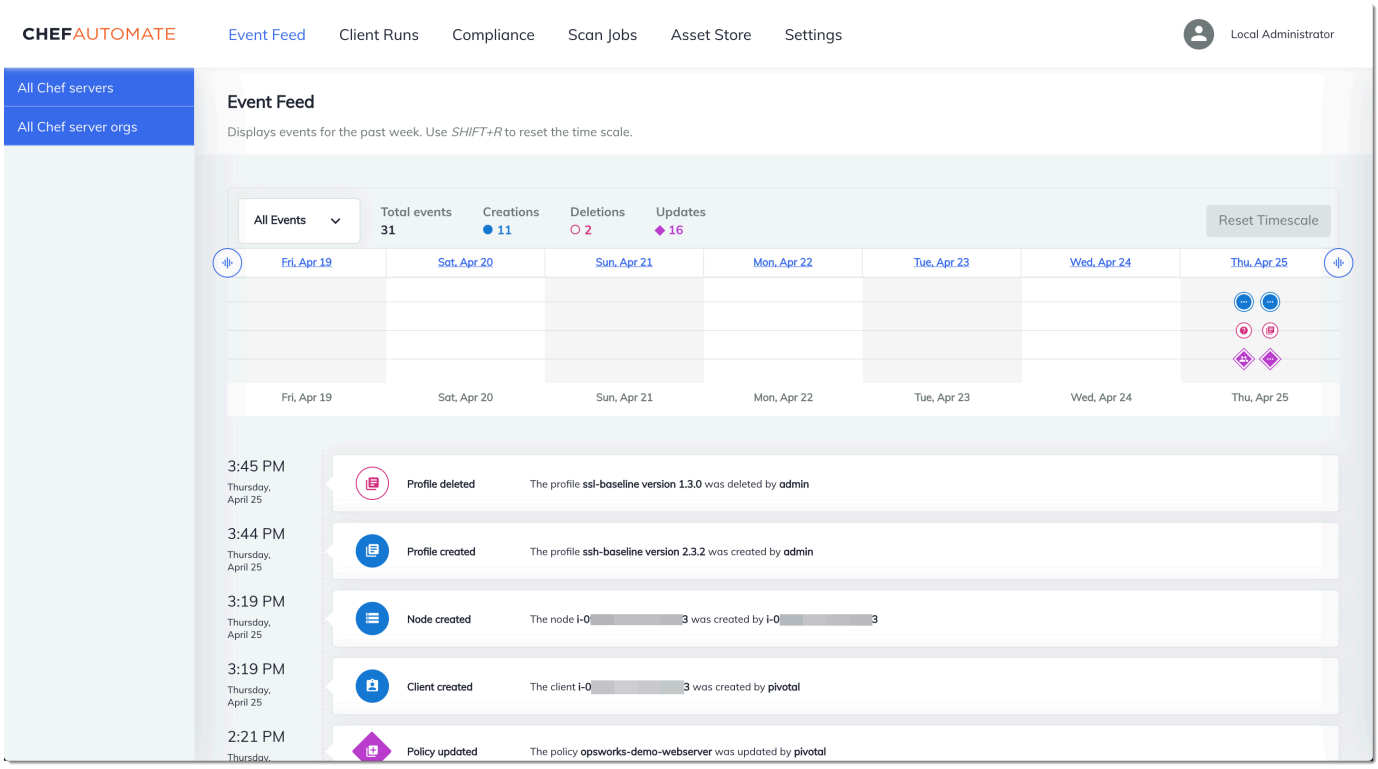
Pour vous connecter au tableau de bord Chef Automate

1. Décompressez et ouvrez les informations d'identification Chef Automate que vous avez téléchargées dans Prérequis. Vous aurez besoin de ces informations d'identification pour vous connecter.

2. Ouvrez la page Properties (Propriétés) de votre serveur Chef.
3. En haut à droite de la page Properties (Propriétés), choisissez Open Chef Automate dashboard (Ouvrir le tableau de bord Chef Automate).
4. Connectez-vous à l'aide des informations d'identification de l'étape 1.



5. Dans le tableau de bord Chef Automate, vous pouvez afficher les informations détaillées sur les nœuds que vous avez amorcés, sur la progression de l'exécution des livres de recettes et les événements, sur le niveau de conformité des nœuds, et plus encore. Pour plus d'informations sur les fonctionnalités du tableau de bord Chef Automate et comment les utiliser, consultez la documentation Chef Automate.



Note

Pour obtenir des informations sur la modification du mot de passe que vous utilisez pour vous connecter au tableau de bord Chef Automate, consultez Réinitialiser les informations d'identification du tableau de bord Chef Automate.

Créez un AWS OpsWorks for Chef Automate serveur en utilisant AWS CloudFormation

Important

AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks for Chef Automate vous permet d'exécuter un serveur Chef Automate dans AWS. Vous pouvez mettre en service un serveur Chef Automate en 15 minutes environ.

À partir du 3 mai 2021, AWS OpsWorks for Chef Automate stocke certains attributs du serveur Chef Automate dans AWS Secrets Manager. Pour de plus amples informations, veuillez consulter Intégration avec AWS Secrets Manager.

La procédure pas à pas suivante vous aide à créer un serveur dans en AWS OpsWorks for Chef Automate créant un stack in AWS CloudFormation.

Rubriques

- Prérequis
- Créer un serveur Chef Automate dans AWS CloudFormation

Prérequis

Avant de créer un nouveau serveur Chef Automate, créez les ressources extérieures à AWS OpsWorks for Chef Automate dont vous aurez besoin pour accéder au serveur Chef et le gérer. Pour plus d'informations, consultez Prérequis dans la section Mise en route de ce guide.

Consultez la section OpsWorks -CM de la référence du modèle de guide de AWS CloudFormation l'utilisateur pour en savoir plus sur les valeurs prises en charge et requises dans le AWS CloudFormation modèle que vous utilisez pour créer votre serveur.

Si vous créez un serveur qui utilise un domaine personnalisé, vous avez besoin d'un domaine, d'un certificat et d'une clé privée personnalisés. Vous devez spécifier des valeurs pour ces trois paramètres dans votre AWS CloudFormation modèle. Pour plus d'informations sur les exigences relatives aux `CustomPrivateKey` paramètres`CustomDomain`,, et`CustomCertificate`,,,,,,, consultez CreateServerle manuel de référence de l'API AWS OpsWorks CM.

Créez un mot de passe pour l'attribut de moteur `CHEF_AUTOMATE_ADMIN_PASSWORD`. La longueur du mot de passe est de huit caractères au minimum et de 32 au maximum. Le mot de passe peut contenir des lettres, des nombres et des caractères spéciaux (`!/@#$%^+=_`). Il doit contenir au moins une lettre en minuscule, une lettre en majuscule, un chiffre et un caractère spécial. Vous spécifiez ce mot de passe dans votre AWS CloudFormation modèle ou en tant que valeur du `CHEF_AUTOMATE_ADMIN_PASSWORD` paramètre lorsque vous créez votre pile.

Générez une paire de clés RSA codée en base64 avant de commencer à créer un serveur Chef Automate dans. AWS CloudFormation La clé publique de la paire est la valeur

deCHEF_AUTOMATE_PIVOTAL_KEY, spécifique à Chef, issue EngineAttributesde l'CreateServerAPI. Cette clé est fournie sous forme de valeur de Parameters dans la AWS CloudFormation console ou dans la create-stack commande du AWS CLI. Pour générer cette clé, nous suggérons les méthodes suivantes.

- Sur les ordinateurs Linux, vous pouvez générer cette clé en exécutant la commande OpenSSL suivante.

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

  Ensuite, exportez dans un fichier la partie clé publique RSA de la paire. La clé publique devient la valeur de CHEF_AUTOMATE_PIVOTAL_KEY.

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

- Sur les ordinateurs Windows, vous pouvez utiliser l'TTYgen utilitaire Pu pour générer une paire de clés RSA codée en base64. Pour plus d'informations, consultez Pu TTYgen - Key Generator for PuTTY sous Windows sur SSH.com.

Créer un serveur Chef Automate dans AWS CloudFormation

Cette section décrit comment utiliser un AWS CloudFormation modèle pour créer une pile qui crée un AWS OpsWorks for Chef Automate serveur. Vous pouvez le faire à l'aide de la AWS CloudFormation console ou du AWS CLI. Un exemple de AWS CloudFormation modèle est à votre disposition pour créer une pile de AWS OpsWorks for Chef Automate serveurs. Veillez à mettre à jour l'exemple de modèle avec votre propre nom de serveur, vos rôles IAM, votre profil d'instance, la description du serveur, le nombre de sauvegardes conservées, les options de maintenance et les balises facultatives. Si votre serveur doit utiliser un domaine personnalisé, vous devez spécifier des valeurs pour les CustomPrivateKey paramètres CustomDomainCustomCertificate, et dans votre AWS CloudFormation modèle. Vous pouvez spécifier les attributs CHEF_AUTOMATE_PIVOTAL_KEY du moteur CHEF_AUTOMATE_ADMIN_PASSWORD et leurs valeurs dans le AWS CloudFormation modèle, ou fournir uniquement les attributs, puis spécifier des valeurs pour les attributs dans l'assistant ou la create-stack commande AWS CloudFormation Create Stack. Pour plus d'informations sur ces attributs, consultez the section called “Créez un serveur Chef Automate dans AWS Management Console” dans la section Mise en route de ce guide.

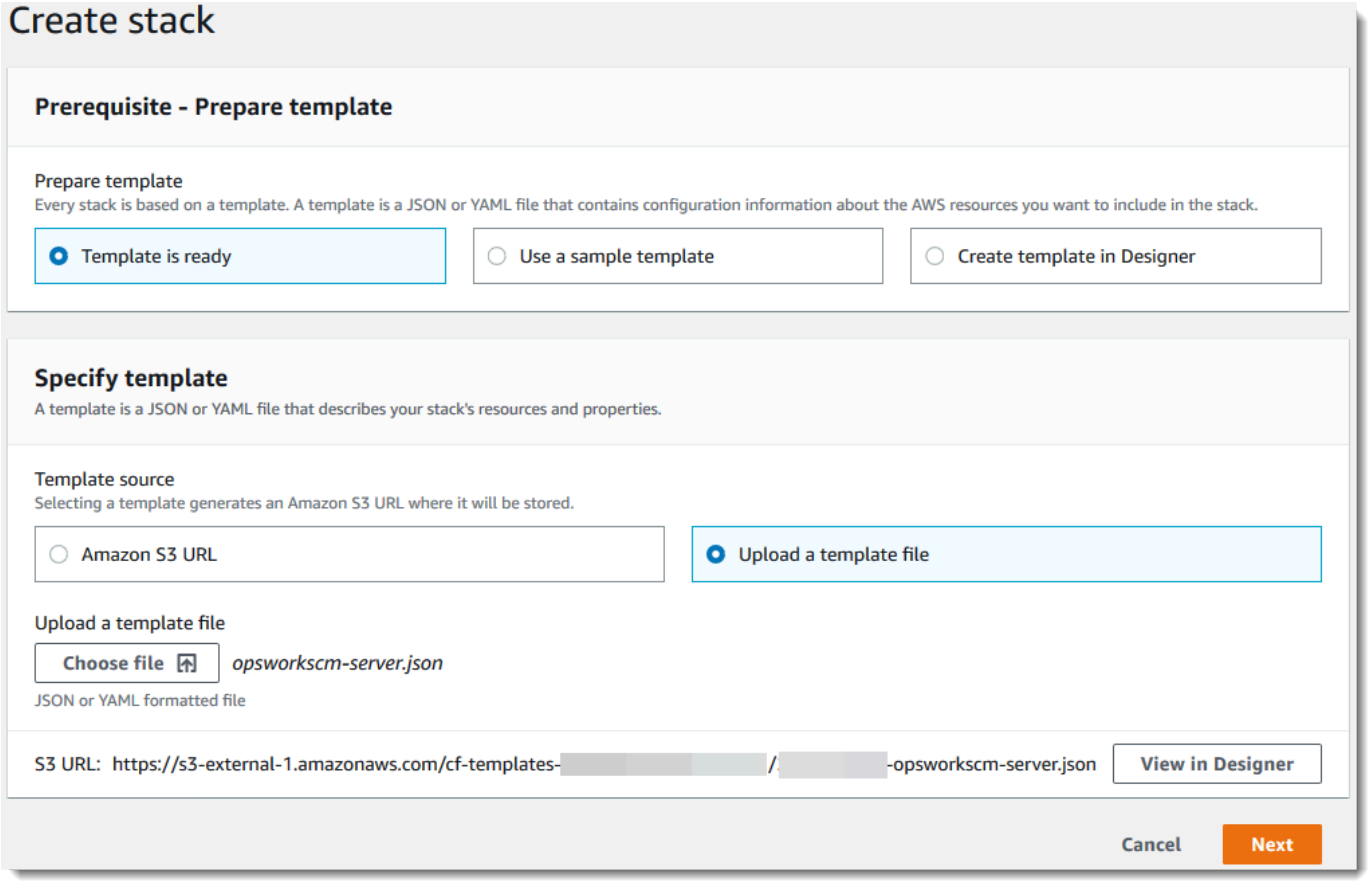
Rubriques

- Créer d'un serveur Chef Automate en utilisant AWS CloudFormation (console)
- Créer un serveur Chef Automate en utilisant AWS CloudFormation (CLI)

Créer d'un serveur Chef Automate en utilisant AWS CloudFormation (console)

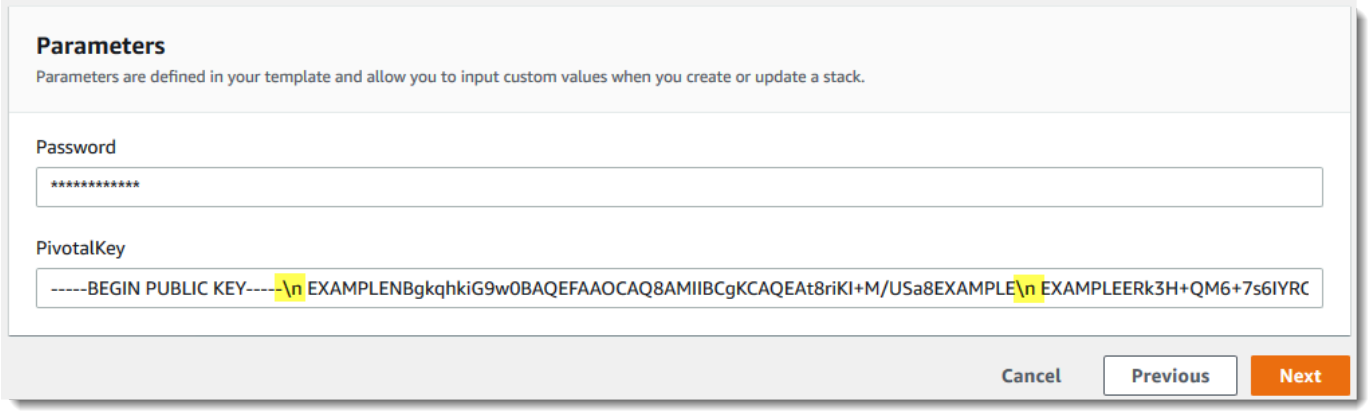
1. Connectez-vous à la AWS CloudFormation console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudformation.
2. Sur la page d' AWS CloudFormation accueil, choisissez Create stack.
3. Dans Prérequis - Préparer le modèle, si vous utilisez le AWS CloudFormation modèle d'exemple, choisissez Le modèle est prêt.
4. Dans Specify template (Spécifier le modèle), choisissez la source de votre modèle. Pour cette procédure pas à pas, choisissez Télécharger un fichier modèle, puis téléchargez un AWS CloudFormation modèle qui crée un serveur Chef Automate. Accédez à votre fichier de modèle, puis choisissez Suivant.

   Un AWS CloudFormation modèle peut être au format YAML ou JSON. Un AWS CloudFormation modèle d'exemple est à votre disposition ; veillez à remplacer les valeurs d'exemple par les vôtres. Vous pouvez utiliser le concepteur de AWS CloudFormation modèles pour créer un nouveau modèle ou valider un modèle existant. Pour plus d'informations sur la procédure, consultez Présentation de l'interface de conception AWS CloudFormation dans le guide de l'utilisateur AWS CloudFormation .



5. Sur la page Specify stack details (Spécifier les détails de la pile), saisissez un nom pour votre pile. Il doit être différent du nom de votre serveur, car il s'agit uniquement d'un nom de pile. Dans la zone Paramètres, collez les valeurs créées dans the section called “Prérequis”. Entrez le mot de passe dans le Mot de passe.

   Collez le contenu du fichier de clé RSA dedans PivotalKey. Dans la AWS CloudFormation console, vous devez ajouter des caractères newline (**\n**) à la fin de chaque ligne de la valeur clé pivot, comme indiqué dans la capture d'écran suivante. Choisissez Suivant.



6. Sur la page Configurer les options de pile, vous pouvez ajouter des balises au serveur que vous créez avec la pile et choisir un rôle IAM pour créer des ressources si vous n'avez pas encore spécifié de rôle IAM à utiliser dans votre modèle. Une fois les options spécifiées, choisissez Suivant. Pour plus d'informations sur les options avancées telles que les déclencheurs d'annulation, consultez la section Configuration des options de AWS CloudFormation pile dans le guide de l'AWS CloudFormation utilisateur.
7. Sur la page Vérification, vérifiez vos choix. Lorsque vous êtes prêt à créer votre pile de serveur, choisissez Create stack (Créer une pile).

   Pendant que vous attendez de AWS CloudFormation créer la pile, consultez l'état de création de la pile. Si la création de la pile échoue, examinez les messages d'erreur affichés dans la console pour vous aider à résoudre le problème. Pour plus d'informations sur la résolution des erreurs dans les piles AWS CloudFormation , consultez Dépannage des erreurs dans le guide de l'utilisateur AWS CloudFormation .

   Lorsque la création du serveur est terminée, votre serveur Chef Automate est disponible sur la page d'accueil d' AWS OpsWorks for Chef Automate , avec un statut en ligne. Générez un nouveau kit de démarrage et les informations d'identification du tableau de bord Chef Automate à partir de la page Propriétés du serveur. Une fois que le serveur est en ligne, le tableau de bord Chef Automate est disponible sur le domaine du serveur, à une URL au format suivant : `https://`*`your_server_name`*`-`*`randomID`*`.`*`region`*`.opsworks-cm.io`.

   > **Note**
   >
   > Si vous avez spécifié un domaine, un certificat et une clé privée personnalisés pour votre serveur, créez une entrée CNAME dans l'outil de gestion DNS de votre entreprise qui associe votre domaine personnalisé au point de terminaison généré AWS OpsWorks for Chef Automate automatiquement pour le serveur. Vous ne pouvez pas gérer le serveur ou vous connecter au tableau de bord Chef Automate pour le serveur tant que vous n'avez pas mappé le point de terminaison généré à votre valeur de domaine personnalisée.
   > Pour obtenir la valeur du point de terminaison généré, exécutez la AWS CLI commande suivante une fois que votre serveur est en ligne :
   >
   > ```
   > aws opsworks describe-servers --server-name server_name
   > ```

Créer un serveur Chef Automate en utilisant AWS CloudFormation (CLI)

Si votre ordinateur local n'exécute pas déjà le AWS CLI, téléchargez-le et installez-le AWS CLI en suivant les instructions d'installation du guide de l'utilisateur de l'interface de ligne de commande AWS. Cette section ne décrit pas tous les paramètres que vous pouvez utiliser avec la commande create-stack. Pour plus d'informations sur les paramètres create-stack, consultez create-stack dans la Référence de l'AWS CLI .

1. Veillez à suivre les étapes requises détaillées dans Prérequis pour créer un serveur AWS OpsWorks for Chef Automate .
2. Créez un rôle de service et un profil d'instance.

   Recherchez et copiez les rôles ARNs de service dans votre compte.

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

   Dans les résultats de la commande `list-roles`, recherchez les entrées de rôle de service et de profil d'instance ressemblant à ce qui suit. Notez le rôle de service et le profil ARNs de l'instance, puis ajoutez-les au AWS CloudFormation modèle que vous utilisez pour créer votre pile de serveurs.

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

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

3. Créez le AWS OpsWorks for Chef Automate serveur en exécutant à nouveau la create-stack commande.

   - Remplacez *stack_name* par le nom de votre pile. Il s'agit du nom de la AWS CloudFormation pile, pas de votre serveur Chef Automate. Le nom du serveur Chef Automate est la valeur de `ServerName` dans votre AWS CloudFormation modèle.
   - *template*Remplacez-le par le chemin d'accès à votre fichier modèle et l'extension *yaml or json* par `.yaml` ou `.json` selon le cas.
   - Les valeurs de `--parameters` correspondent à celles EngineAttributesde l'CreateServerAPI. Pour Chef, les attributs de moteur fournis par l'utilisateur pour créer un serveur sont `CHEF_AUTOMATE_PIVOTAL_KEY`, une clé publique RSA encodée en base64 générée à l'aide des utilitaires décrits dans the section called “Prérequis”, et `CHEF_AUTOMATE_ADMIN_PASSWORD`, un mot de passe de 8 à 32 caractères créé par vos soins. Pour en savoir plus sur `CHEF_AUTOMATE_ADMIN_PASSWORD`, consultez Créez un serveur Chef Automate à l'aide du AWS CLI. Vous pouvez fournir un pointeur vers le fichier PEM contenant votre clé pivotale comme valeur du paramètre `PivotalKey`, tel qu'illustré dans l'exemple. Si les valeurs pour `CHEF_AUTOMATE_ADMIN_PASSWORD` et ne `CHEF_AUTOMATE_PIVOTAL_KEY` sont pas spécifiées dans votre modèle, vous devez les fournir dans votre AWS CLI commande.

```
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"
```

Voici un exemple qui inclut des exemples de valeurs pour les attributs `CHEF_AUTOMATE_PIVOTAL_KEY` et `CHEF_AUTOMATE_ADMIN_PASSWORD`. Exécutez une commande similaire si vous n'avez pas spécifié de valeurs pour ces attributs dans votre AWS CloudFormation modèle.

```
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. Lorsque la création de la pile est terminée, ouvrez la page Propriétés du nouveau serveur dans la AWS OpsWorks for Chef Automate console et téléchargez un kit de démarrage. Le téléchargement d'un nouveau kit de démarrage réinitialise le mot de passe administrateur du tableau de bord Chef Automate.
5. Si votre serveur utilise un domaine, un certificat et une clé privée personnalisés, suivez les étapes de configuration de `knife.rb` dans (Facultatif) Configurer `knife` pour utiliser un domaine personnalisé, puis passez à l'étape 7.
6. Pour utiliser des commandes `knife` sur le nouveau serveur, mettez à jour les paramètres du fichier de configuration `knife.rb` Chef. Un exemple de fichier `knife.rb` est inclus avec le kit de démarrage. L'exemple suivant montre comment configurer `knife.rb` sur un serveur qui n'utilise pas de domaine personnalisé. Si vous utilisez un domaine personnalisé, consultez (Facultatif) Configurer `knife` pour utiliser un domaine personnalisé pour obtenir les instructions de configuration de `knife`.

   - Remplacez *ENDPOINT* par la valeur du point de terminaison du serveur. Cela fait partie de la sortie de l'opération de création de pile. Vous pouvez supprimer le point de terminaison en exécutant la commande suivante.

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

- Remplacez *key_pair_file.pem* dans la `client_key` configuration par le nom du fichier PEM qui contient celui `CHEF_AUTOMATE_PIVOTAL_KEY` que vous avez utilisé pour créer votre serveur.

```
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. Lorsque le processus de création du serveur est terminé, accédez à the section called “Fin de la configuration et chargement des livres de recettes”. Si la création de la pile échoue, examinez les messages d'erreur affichés dans la console pour vous aider à résoudre le problème. Pour plus d'informations sur la résolution des erreurs dans les AWS CloudFormation piles, consultez la section Résolution des erreurs dans le guide de l'AWS CloudFormation utilisateur.

Mettre à jour un AWS OpsWorks for Chef Automate serveur pour utiliser un domaine personnalisé

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section décrit comment mettre à jour un AWS OpsWorks for Chef Automate serveur existant pour utiliser un domaine et un certificat personnalisés en utilisant une sauvegarde du serveur pour créer un nouveau serveur. Essentiellement, vous copiez un serveur AWS OpsWorks for Chef

Automate 2.0 existant en créant un nouveau serveur à partir d'une sauvegarde, puis en configurant le nouveau serveur pour utiliser un domaine, un certificat et une clé privée personnalisés.

Rubriques

- Prérequis
- Limites
- Mise à jour d'un serveur pour utiliser un domaine personnalisé
- consultez aussi

Prérequis

Les conditions suivantes sont requises pour mettre à jour un AWS OpsWorks for Chef Automate serveur existant afin d'utiliser un domaine et un certificat personnalisés.

- Le serveur que vous souhaitez mettre à jour (ou copier) doit exécuter Chef Automate 2.0.
- Choisissez la sauvegarde à utiliser pour créer un nouveau serveur. Vous devez disposer d'au moins une sauvegarde du serveur que vous souhaitez mettre à jour. Pour plus d'informations sur les sauvegardes dans AWS OpsWorks for Chef Automate, consultezSauvegarder un AWS OpsWorks for Chef Automate serveur.
- Préparez le rôle de service et le profil d'instance ARNs que vous avez utilisés pour créer le serveur existant qui est la source de votre sauvegarde.
- Assurez-vous que vous exécutez la version la plus récente de l' AWS CLI. Pour plus d'informations sur la mise à jour de vos AWS CLI outils, consultez le guide de AWS CLI l'utilisateur de l'interface de ligne de commande AWS sur l'installation.

Limites

Lorsque vous mettez à jour un serveur existant en créant un nouveau serveur à partir d'une sauvegarde, le nouveau serveur ne peut pas être exactement le même que le AWS OpsWorks for Chef Automate serveur existant.

- Vous ne pouvez effectuer cette procédure qu'en utilisant le AWS CLI ou l'un des AWS SDKs. Vous ne pouvez pas créer un nouveau serveur à partir d'une sauvegarde avec AWS Management Console.

- Le nouveau serveur ne peut pas utiliser le même nom que le serveur existant au sein d'un compte et d'une région AWS. Le nom doit être différent de celui du serveur existant que vous avez utilisé comme source de la sauvegarde.
- Les nœuds attachés au serveur existant ne sont pas gérés par le nouveau serveur. Vous devez effectuer l'une des opérations suivantes.
  - Attachez des nœuds différents, car des nœuds ne peuvent pas être gérés par plusieurs serveurs Chef Automate.
  - Migrez les nœuds à partir du serveur existant (source de la sauvegarde) vers le nouveau serveur et le nouveau point de terminaison de domaine personnalisé. Pour plus d'informations sur la migration des nœuds, consultez la documentation Chef.

Mise à jour d'un serveur pour utiliser un domaine personnalisé

Pour mettre à jour un serveur Chef Automate 2.0 existant, vous en faites une copie en exécutant la commande `create-server`, et en ajoutant des paramètres pour spécifier une sauvegarde, un domaine personnalisé, un certificat personnalisé et une clé privée personnalisée.

1. Si aucun rôle de service ou profil d'instance n' ARNs est disponible à spécifier dans votre `create-server` commande, suivez les étapes 1 Créez un serveur Chef Automate à l'aide du AWS CLI à 5 pour créer un rôle de service ou un profil d'instance que vous pouvez utiliser.
2. Si vous ne l'avez pas déjà fait, recherchez la sauvegarde du serveur Chef Automate 2.0 existant sur lequel vous souhaitez baser un nouveau serveur avec un domaine personnalisé. Exécutez la commande suivante pour afficher les informations relatives à toutes les AWS OpsWorks for Chef Automate sauvegardes de votre compte et d'une région. Veillez à notez l'ID de la sauvegarde que vous souhaitez utiliser.

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

3. Créez le AWS OpsWorks for Chef Automate serveur en exécutant la `create-server` commande.

   - La valeur de `--engine` est `ChefAutomate`, `--engine-model` a pour valeur `Single` et `--engine-version` a pour valeur `12`.
   - Le nom du serveur doit être unique dans votre AWS compte, dans chaque région. Les noms de serveur doivent commencer par une lettre. Ensuite, vous pouvez utiliser des lettres, des chiffres ou des tirets (-), jusqu'à un maximum de 40 caractères.

- Utilisez l'ARN du profil d'instance et l'ARN du rôle de service de l'étape 1.
- Les types d'instance valides sont `m5.large`, `r5.xlarge` et `r5.2xlarge`. Pour plus d'informations sur les spécifications de ces types d'instances, consultez la section Types d'instances du guide de EC2 l'utilisateur Amazon.
- Le paramètre `--engine-attributes` est facultatif ; si vous ne spécifiez pas l'une des valeurs ou les deux, le processus de création de serveur les génère pour vous. Si vous ajoutez `--engine-attributes`, spécifiez la valeur `CHEF_AUTOMATE_PIVOTAL_KEY` que vous avez générée à l'étape 2, une valeur `CHEF_AUTOMATE_ADMIN_PASSWORD`, ou les deux.

  Si vous ne définissez pas de valeur pour `CHEF_AUTOMATE_ADMIN_PASSWORD`, un mot de passe est généré et inclus dans la réponse `create-server`. Vous pouvez également télécharger à nouveau le kit de démarrage dans la console, ce qui permet de générer à nouveau ce mot de passe. La longueur du mot de passe est de huit caractères au minimum et de 32 au maximum. Le mot de passe peut contenir des lettres, des nombres et des caractères spéciaux (`!/@#$%^+=_`). Il doit contenir au moins une lettre en minuscule, une lettre en majuscule, un chiffre et un caractère spécial.
- Une paire de clés SSH est facultative, mais elle peut vous aider à vous connecter à votre serveur Chef Automate si vous devez réinitialiser le mot de passe de l'administrateur du tableau de bord Chef Automate. Pour plus d'informations sur la création d'une paire de clés SSH, consultez Amazon EC2 Key Pairs dans le guide de l' EC2 utilisateur Amazon.
- Pour utiliser un domaine personnalisé, ajoutez les paramètres suivants à votre commande. Sinon, le processus de création de serveur Chef Automate génère automatiquement un point de terminaison pour vous. Les trois paramètres sont requis pour configurer un domaine personnalisé. Pour plus d'informations sur les exigences supplémentaires relatives à l'utilisation de ces paramètres, consultez CreateServerla référence de l'API AWS OpsWorks CM.
  - `--custom-domain` - Point de terminaison public facultatif d'un serveur, tel que `https://aws.my-company.com`.
  - `--custom-certificate` - Un certificat HTTPS au format PEM. La valeur peut être un seul certificat auto-signé ou une chaîne de certificats.
  - `--custom-private-key` - Une clé privée au format PEM pour se connecter au serveur à l'aide de HTTPS. La clé privée ne doit pas être chiffrée ; elle ne peut pas être protégée par un mot de passe ou une phrase passe.
- Une maintenance hebdomadaire du système est obligatoire. Vous devez spécifier des valeurs valides au format suivant : `DDD:HH:MM`. L'heure doit être exprimée en heure UTC

(temps universel coordonné). Si vous ne spécifiez aucune valeur pour `--preferred-maintenance-window`, la valeur par défaut est un intervalle aléatoire d'une heure le mardi, le mercredi ou le vendredi.

- Les valeurs valides pour `--preferred-backup-window` doivent être spécifiées dans l'un des formats suivants : `HH:MM` pour les sauvegardes quotidiennes ou `DDD:HH:MM` pour les sauvegardes hebdomadaires. L'heure indiquée est en heure UTC. La valeur par défaut est une heure de début quotidienne aléatoire. Pour annuler les sauvegardes quotidiennes, ajoutez le paramètre `--disable-automated-backup`.
- Pour`--security-group-ids`, entrez un ou plusieurs groupes de sécurité IDs, séparés par un espace.
- Pour `--subnet-ids`, entrez un ID de sous-réseau.
- Pour `--backup-id`, entrez l'ID de la sauvegarde que vous avez copiée à l'étape 2.

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

L'exemple suivant crée un serveur Chef Automate qui utilise un domaine personnalisé.

```
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 prend environ 15 minutes pour créer un nouveau serveur. Dans la sortie de la commande `create-server`, copiez la valeur de l'attribut `Endpoint`. Voici un exemple.

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

   Ne supprimez pas la sortie de la commande `create-server` ou ne fermez pas votre session shell, car la sortie peut contenir des informations importantes qui ne seront plus réaffichées. Pour obtenir les mots de passe et le kit de démarrage à partir des résultats de la commande `create-server`, passez à l'étape suivante.

5. Si vous avez choisi de AWS OpsWorks for Chef Automate générer une clé et un mot de passe pour vous, vous pouvez les extraire dans des formats utilisables à partir des `create-server` résultats à l'aide d'un processeur JSON tel que jq. Une fois que vous avez installé jq, vous pouvez exécuter les commandes suivantes pour extraire la clé centrale, le mot de passe administrateur du tableau de bord Chef Automate et le kit de démarrage. Si vous n'avez pas fourni votre propre clé centrale et votre mot de passe à l'étape 3, veillez à sauvegarder la clé centrale et le mot de passe administrateur extraits dans des emplacements sécurisés, mais pratiques.

   ```
   #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. Si vous n'avez pas extrait le kit de démarrage à partir des résultats des `create-server` commandes, vous pouvez éventuellement télécharger un nouveau kit de démarrage depuis la page Propriétés du serveur dans la AWS OpsWorks for Chef Automate console. Le téléchargement d'un nouveau kit de démarrage réinitialise le mot de passe administrateur du tableau de bord Chef Automate.
7. Créez une entrée CNAME dans l'outil de gestion DNS de votre entreprise pour faire pointer votre domaine personnalisé vers le point de AWS OpsWorks for Chef Automate terminaison que vous avez copié à l'étape 4. Vous ne pouvez pas accéder ou vous connecter au serveur tant que vous n'avez pas effectué cette étape.
8. Lorsque le processus de création du serveur est terminé, accédez à the section called “Fin de la configuration et chargement des livres de recettes”.

consultez aussi

- Créez un serveur Chef Automate à l'aide du AWS CLI
- Restaurer un AWS OpsWorks for Chef Automate serveur à partir d'une sauvegarde
- CreateServerdans la référence de l'API AWS OpsWorks CM
- create-server dans la référence de commande de l'AWS CLI

Régénérer le kit de démarrage d'un serveur AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Le kit de démarrage AWS OpsWorks for Chef Automate contient un fichier README avec des exemples, un fichier de `knife.rb` configuration et une clé privée pour l'utilisateur principal, ou pivot. Une nouvelle paire de clés est générée, et l'ancienne clé est réinitialisée, chaque fois que vous

téléchargez le kit de démarrage. Vous pouvez régénérer le kit de démarrage d'un AWS OpsWorks for Chef Automate serveur de deux manières différentes :

- Dans la AWS OpsWorks console, dans le menu Actions de la page de détails d'un AWS OpsWorks for Chef Automate serveur. Vous êtes invité à confirmer si vous souhaitez régénérer et réinitialiser l'ancienne clé pivot.
- En exécutant des commandes dans le AWS CLI.

Pour plus d'informations sur l'utilisation du kit de démarrage, voirConfiguration d'un serveur Chef à l'aide du Starter Kit.

Régénérez le kit de AWS OpsWorks for Chef Automate démarrage avec AWS CLI

> **Note**
>
> Lorsque vous régénérez le kit de démarrage, vous régénérez et réinitialisez également la paire de clés d'authentification pour votre serveur Chef Automate, et vous supprimez la paire de clés actuelle.

Régénérez le kit de démarrage en exécutant la update-server-engine-attributescommande.
Au AWS CLI cours d'une session, exécutez la commande suivante. Spécifiez le nom de votre serveur comme valeur de`--server-name`. Pour définir votre propre clé publique comme valeur de`CHEF_AUTOMATE_PIVOTAL_KEY`, spécifiez la valeur de la clé publique dans`--attribute-value`. Dans le cas contraire, définissez `--attribute-value` la valeur sur null.

```
aws opsworks-cm update-server-engine-attributes \
   --server-name server_name \
   --attribute-name "CHEF_AUTOMATE_PIVOTAL_KEY" \
   --attribute-value your_public_key
```

La commande suivante est un exemple qui spécifie une valeur de clé publique que l'administrateur du serveur souhaite utiliser.

```
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-----"
```

La commande suivante est un exemple qui permet de AWS OpsWorks for Chef Automate régénérer la clé publique.

```
aws opsworks-cm update-server-engine-attributes \
   --server-name your-test-server \
   --attribute-name "CHEF_AUTOMATE_PIVOTAL_KEY" \
   --attribute-value null
```

Le résultat de cette commande est constitué d'informations sur le serveur et d'un fichier ZIP codé en base64. Le fichier ZIP contient un kit de démarrage Chef, qui comprend un fichier README, un fichier de configuration et la clé privée RSA requise. Enregistrez ce fichier, décompressez-le, puis accédez au répertoire dans lequel vous avez décompressé le contenu du fichier. À partir de ce répertoire, vous pouvez exécuter `knife` des commandes.

Utilisation des balises sur les AWS OpsWorks for Chef Automate ressources

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les balises sont des mots ou des expressions qui jouent le rôle de métadonnées pour identifier et organiser vos ressources AWS. Dans AWS OpsWorks for Chef Automate, une ressource peut comporter jusqu'à 50 balises appliquées par l'utilisateur. Chaque balise est constituée d'une clé et d'une valeur facultative. Vous pouvez appliquer des balises aux ressources suivantes dans AWS OpsWorks for Chef Automate :

- AWS OpsWorks for Chef Automate serveurs
- Sauvegardes de AWS OpsWorks for Chef Automate serveurs

Les balises sur les AWS ressources peuvent vous aider à suivre les coûts, à contrôler l'accès aux ressources, à regrouper les ressources pour automatiser les tâches ou à organiser les ressources par objectif ou par étape du cycle de vie. Pour de plus amples informations sur les avantages des balises, veuillez consulter la page relative aux stratégies de balisage AWS dans AWS Answers et Utilisation des balises de répartition des coûts dans le Guide de l'utilisateur AWS Billing and Cost Management .

Pour utiliser des balises afin de contrôler l'accès aux AWS OpsWorks for Chef Automate serveurs ou aux sauvegardes, vous devez créer ou modifier des déclarations de politique dans AWS Identity and Access Management (IAM). Pour de plus amples informations, veuillez consulter Contrôle de l'accès aux ressources AWS à l'aide des balises de ressources dans le Guide de l'utilisateur AWS Identity and Access Management .

Lorsque vous appliquez des balises à un AWS OpsWorks for Chef Automate serveur, celles-ci sont également appliquées aux sauvegardes du serveur, au compartiment Amazon S3 qui stocke les sauvegardes, à l' EC2 instance Amazon du serveur, aux secrets du serveur qui y sont stockés et à AWS Secrets Manager l'adresse IP élastique utilisée par le serveur. Les balises ne sont pas propagées à la AWS CloudFormation pile AWS OpsWorks utilisée pour créer votre serveur.

Rubriques

- Comment fonctionnent les tags dans AWS OpsWorks for Chef Automate
- Ajouter et gérer des balises dans AWS OpsWorks for Chef Automate (console)
- Ajouter et gérer des balises dans AWS OpsWorks for Chef Automate (CLI)
- consultez aussi

Comment fonctionnent les tags dans AWS OpsWorks for Chef Automate

Dans cette version, vous pouvez ajouter et gérer des balises à l'aide de l'API AWS OpsWorks CM ou de l' AWS Management Console. AWS OpsWorks CM tente également d'ajouter des balises que vous ajoutez à un serveur aux AWS ressources associées au serveur, notamment l' EC2 instance, les secrets dans Secrets Manager, l'adresse IP Elastic, le groupe de sécurité, le compartiment S3 et les sauvegardes. Le tableau suivant donne une vue d'ensemble de la façon dont vous ajoutez et gérez des balises dans AWS OpsWorks for Chef Automate.

Action	Ce qu'il faut utiliser
Ajoutez des balises à un nouveau AWS OpsWorks for Chef Automate serveur ou à une sauvegarde que vous créez manuellement.	• Choisissez Créer un serveur Chef Automate et ajoutez des balises sur la page Configurer les paramètres avancés. • Choisissez Créer une sauvegarde sur la page Sauvegardes d'un serveur existant et ajoutez des balises sur la page Créer une sauvegarde de votre serveur Chef Automate 2 . • Ajoutez un paramètre `Tags` aux commandes `CreateServer` ou `CreateBackup` .
Affichez les balises sur une ressource.	• Sur la page de détails de votre serveur, choisissez Balises dans le panneau de navigation. • Sur la page Sauvegardes de votre serveur, sélectionnez une sauvegarde, puis choisisse z Modifier la sauvegarde. • Exécutez la commande `ListTagsForResource` .
Ajoutez des balises à un AWS OpsWorks for Chef Automate serveur existant ou à une sauvegarde, que la sauvegarde ait été créée manuellement ou automatiquement.	• Sur la page de détails de votre serveur, choisissez Balises dans le panneau de navigation, puis choisissez Modifier. • Sur la page Sauvegardes de votre serveur, sélectionnez une sauvegarde, puis choisisse z Modifier la sauvegarde. • Exécutez la commande `TagResource` .
Supprimez des balises d'une ressource.	• Sur la page de détails de votre serveur, choisissez Balises dans le panneau de navigation, puis choisissez Modifier. Choisissez X en regard des balises que vous souhaitez supprimer.

Action	Ce qu'il faut utiliser
	• Sur la page Sauvegardes de votre serveur, sélectionnez une sauvegarde, puis choisissez Modifier la sauvegarde. Choisissez X en regard des balises que vous souhaitez supprimer. • Exécutez la commande `UntagResource`.

Les réponses `DescribeServers` et `DescribeBackups` n'incluent pas d'informations sur les balises. Pour afficher les balises, utilisez l'API `ListTagsForResource`.

Ajouter et gérer des balises dans AWS OpsWorks for Chef Automate (console)

Les procédures de cette section sont effectuées dans l' AWS Management Console.

Si vous ajoutez des balises, une clé de balise ne peut pas être vide. La clé peut comporter 127 caractères maximum et ne peut pas contenir uniquement des lettres Unicode, des chiffres ou des séparateurs, ni les caractères spéciaux suivants : `+ - = . _ : / @`. Une valeur de balise est facultative. Vous pouvez ajouter une balise ayant une clé, mais aucune valeur. La valeur peut comporter 255 caractères maximum et ne peut contenir que des lettres Unicode, des chiffres, des séparateurs ou les caractères spéciaux suivants : `+ - = . _ : / @`.

Rubriques

- Ajouter des balises à un nouveau AWS OpsWorks for Chef Automate serveur (console)
- Ajouter des balises à une nouvelle sauvegarde (Console)
- Ajouter ou afficher des balises sur un serveur existant (Console)
- Ajouter ou afficher des balises sur une sauvegarde existante (Console)
- Supprimer les balises d'un serveur (Console)
- Supprimer les balises d'une sauvegarde (Console)

Ajouter des balises à un nouveau AWS OpsWorks for Chef Automate serveur (console)

1. Assurez-vous de remplir toutes les conditions préalables à la création d'un AWS OpsWorks for Chef Automate serveur.
2. Suivez les étapes 1 à 10 dans Création d'un serveur Chef Automate.
3. Après avoir spécifié les paramètres de sauvegarde automatisée, ajoutez des balises dans la zone Balises de la page Configurer les paramètres avancés. Vous pouvez ajouter un maximum de 50 balises. Lorsque vous avez fini d'ajouter des balises, choisissez Suivant.
4. Passez à l'étape 13 de Création d'un serveur Chef Automate, et passez en revue les paramètres que vous avez choisis pour le nouveau serveur.

Ajouter des balises à une nouvelle sauvegarde (Console)

1. Sur la page d' AWS OpsWorks for Chef Automate accueil, choisissez un serveur Chef Automate existant.
2. Dans la page de détails du serveur, choisissez Sauvegardes dans le panneau de navigation.
3. Sur la page Sauvegardes choisissez Créer une sauvegarde.
4. Ajouter des balises. Une fois que vous avez fini d'ajouter des règles, choisissez Créer.

Ajouter ou afficher des balises sur un serveur existant (Console)

1. Sur la page d' AWS OpsWorks for Chef Automate accueil, choisissez un serveur Chef Automate existant pour ouvrir sa page de détails.
2. Choisissez Balises dans le panneau de navigation ou, en bas de la page de détails, choisissez Afficher toutes les balises.
3. Sur la page Balises, choisissez Modifier.
4. Ajoutez ou modifiez des balises sur le serveur. Lorsque vous avez terminé, choisissez Save (Enregistrer)

> **Note**
>
> Sachez que la modification des balises sur votre serveur Chef Automate modifie également les balises des ressources associées au serveur, telles que l' EC2 instance, l'adresse IP élastique, le groupe de sécurité, le compartiment S3 et les sauvegardes.

Ajouter ou afficher des balises sur une sauvegarde existante (Console)

1. Sur la page d' AWS OpsWorks for Chef Automate accueil, choisissez un serveur Chef Automate existant pour ouvrir sa page de détails.
2. Choisissez Sauvegardes dans le panneau de navigation ou, dans la zone Sauvegardes récentes de la page de détails, choisissez Afficher toutes les sauvegardes.
3. Sur la page Sauvegardes, choisissez une sauvegarde à gérer, puis choisissez Modifier la sauvegarde.
4. Ajoutez ou modifiez des balises sur la sauvegarde. Choisissez Mettre à jour une fois que vous avez terminé.

Supprimer les balises d'un serveur (Console)

1. Sur la page d' AWS OpsWorks for Chef Automate accueil, choisissez un serveur Chef Automate existant pour ouvrir sa page de détails.
2. Choisissez Balises dans le panneau de navigation ou, en bas de la page de détails, choisissez Afficher toutes les balises.
3. Sur la page Balises, choisissez Modifier.
4. Choisissez X en regard d'une balise pour la supprimer. Lorsque vous avez terminé, choisissez Save (Enregistrer)

> **Note**
>
> Sachez que la modification des balises sur votre serveur Chef Automate modifie également les balises des ressources associées au serveur, telles que l' EC2 instance, l'adresse IP élastique, le groupe de sécurité, le compartiment S3 et les sauvegardes.

Supprimer les balises d'une sauvegarde (Console)

1. Sur la page d' AWS OpsWorks for Chef Automate accueil, choisissez un serveur Chef Automate existant pour ouvrir sa page de détails.
2. Choisissez Sauvegardes dans le panneau de navigation ou, dans la zone Sauvegardes récentes de la page de détails, choisissez Afficher toutes les sauvegardes.
3. Sur la page Sauvegardes, choisissez une sauvegarde à gérer, puis choisissez Modifier la sauvegarde.
4. Choisissez X en regard d'une balise pour la supprimer. Choisissez Mettre à jour une fois que vous avez terminé.

Ajouter et gérer des balises dans AWS OpsWorks for Chef Automate (CLI)

Les procédures de cette section sont effectuées dans l' AWS CLI. Assurez-vous que vous utilisez la dernière version du AWS CLI avant de commencer à travailler avec les balises. Pour plus d'informations sur l'installation ou la mise à jour du AWS CLI, voir Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Si vous ajoutez des balises, une clé de balise ne peut pas être vide. La clé peut comporter 127 caractères maximum et ne peut pas contenir uniquement des lettres Unicode, des chiffres ou des séparateurs, ni les caractères spéciaux suivants : `+ - = . _ : / @`. Une valeur de balise est facultative. Vous pouvez ajouter une balise ayant une clé, mais aucune valeur. La valeur peut comporter 255 caractères maximum et ne peut contenir que des lettres Unicode, des chiffres, des séparateurs ou les caractères spéciaux suivants : `+ - = . _ : / @`.

Rubriques

- Ajouter des balises à un nouveau AWS OpsWorks for Chef Automate serveur (CLI)
- Ajouter des balises à une nouvelle sauvegarde (interface de ligne de commande)
- Ajouter des balises à des serveurs ou sauvegardes existants (Interface de ligne de commande)
- Liste des balises de ressource
- Supprimer des balises d'une ressource

Ajouter des balises à un nouveau AWS OpsWorks for Chef Automate serveur (CLI)

Vous pouvez utiliser le AWS CLI pour ajouter des balises lorsque vous créez un AWS OpsWorks for Chef Automate serveur. Cette procédure ne décrit pas en détail la création d'un serveur. Pour obtenir

des informations détaillées sur la création d'un AWS OpsWorks for Chef Automate serveur à l' AWS CLI aide Créez un serveur Chef Automate à l'aide du AWS CLI de ce guide. Vous pouvez ajouter jusqu'à 50 balises à un serveur.

1. Assurez-vous de remplir toutes les conditions préalables à la création d'un AWS OpsWorks for Chef Automate serveur.
2. Effectuez les étapes 1 à 5 de Créez un serveur Chef Automate à l'aide du AWS CLI.
3. Pour l'étape 6, lorsque vous exécutez la commande `create-server`, ajoutez le paramètre `--tags` à la commande, comme indiqué dans l'exemple suivant.

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

   Voici un exemple montrant uniquement la partie correspondant aux balises de la commande `create-server`.

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

4. Suivez les étapes restantes fournies dans Créez un serveur Chef Automate à l'aide du AWS CLI. Pour vérifier que vos balises ont été ajoutées au nouveau serveur, suivez les étapes décrites dans Liste des balises de ressource de la présente rubrique.

Ajouter des balises à une nouvelle sauvegarde (interface de ligne de commande)

Vous pouvez utiliser le AWS CLI pour ajouter des balises lorsque vous créez une nouvelle sauvegarde manuelle d'un AWS OpsWorks for Chef Automate serveur. Cette procédure ne décrit pas en détail la création d'une sauvegarde manuelle. Pour obtenir des informations détaillées sur la création d'une sauvegarde manuelle, reportez-vous à « Pour effectuer une sauvegarde manuelle dans le AWS CLI » dansSauvegarder un AWS OpsWorks for Chef Automate serveur. Vous pouvez ajouter jusqu'à 50 balises à une sauvegarde. Si un serveur possède des balises, les nouvelles sauvegardes sont automatiquement balisées avec les balises du serveur.

Par défaut, lorsque vous créez un nouveau AWS OpsWorks for Chef Automate serveur, les sauvegardes automatiques sont activées. Vous pouvez ajouter des balises à une sauvegarde automatisée en exécutant la commande `tag-resource` décrite dans Ajouter des balises à des serveurs ou sauvegardes existants (Interface de ligne de commande) de la présente rubrique.

- Pour ajouter des balises à une sauvegarde manuelle lors de la création de la sauvegarde, exécutez la commande suivante. Seule la partie de la commande correspondant aux balises est affichée. Pour obtenir un exemple de la commande `create-backup` complète, veuillez consulter Pour effectuer une sauvegarde manuelle dans l' AWS CLI" dans Sauvegarder un AWS OpsWorks for Chef Automate serveur.

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

L'exemple suivant montre uniquement la partie de la commande `create-backup` correspondant aux balises.

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

Ajouter des balises à des serveurs ou sauvegardes existants (Interface de ligne de commande)

Vous pouvez exécuter la commande `tag-resource` pour ajouter des balises à des serveurs ou des sauvegardes AWS OpsWorks for Chef Automate existants (que les sauvegardes aient été créées automatiquement ou manuellement). Spécifiez l'ARN (Amazon Resource Name) d'une ressource cible pour y ajouter des balises.

1. Pour obtenir l'ARN de la ressource à laquelle vous souhaitez appliquer des balises :

   - Pour un serveur, exécutez `describe-servers --server-name` *`server_name`*. Les résultats de la commande indiquent l'ARN du serveur.
   - Pour une sauvegarde, exécutez `describe-backups --backup-id` *`backup_ID`*. Les résultats de la commande montrent l'ARN de la sauvegarde. Vous pouvez également exécuter `describe-backups --server-name` *`server_name`* pour afficher des informations sur toutes les sauvegardes d'un AWS OpsWorks for Chef Automate serveur spécifique.

   L'exemple suivant montre uniquement la valeur `ServerArn` dans les résultats d'une commande `describe-servers --server-name opsworks-cm-test`. La valeur `ServerArn` est ajoutée à une commande `tag-resource` pour ajouter des balises au serveur.

```
{
```

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

2. Exécutez la commande `tag-resource` avec l'ARN que vous avez renvoyé à l'étape 1.

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

Voici un exemple.

```
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. Pour vérifier que les balises ont été ajoutées avec succès, passez à la procédure suivante, Liste des balises de ressource.

Liste des balises de ressource

Vous pouvez exécuter la `list-tags-for-resource` commande pour afficher les balises associées aux AWS OpsWorks for Chef Automate serveurs ou aux sauvegardes. Spécifiez l'ARN d'une ressource cible pour afficher ses balises.

1. Pour obtenir l'ARN de la ressource pour laquelle vous souhaitez répertorier les balises :

   - Pour un serveur, exécutez `describe-servers --server-name` *`server_name`*. Les résultats de la commande indiquent l'ARN du serveur.
   - Pour une sauvegarde, exécutez `describe-backups --backup-id` *`backup_ID`*. Les résultats de la commande montrent l'ARN de la sauvegarde. Vous pouvez également exécuter `describe-backups --server-name` *`server_name`* pour afficher des informations sur toutes les sauvegardes d'un AWS OpsWorks for Chef Automate serveur spécifique.

2. Exécutez la commande `list-tags-for-resource` avec l'ARN que vous avez renvoyé à l'étape 1.

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

Voici un exemple.

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

Si la ressource comporte des balises, la commande renvoie des résultats similaires au résultat suivant.

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

Supprimer des balises d'une ressource

Vous pouvez exécuter la commande `untag-resource` pour supprimer des balises des serveurs ou des sauvegardes AWS OpsWorks for Chef Automate . Si la ressource est supprimée, les balises de la ressource sont également supprimées. Spécifiez l'ARN (Amazon Resource Name) d'une ressource cible pour en supprimer les balises.

1. Pour obtenir l'ARN de la ressource pour laquelle vous souhaitez supprimer des balises :

   - Pour un serveur, exécutez `describe-servers --server-name` *`server_name`*. Les résultats de la commande indiquent l'ARN du serveur.
   - Pour une sauvegarde, exécutez `describe-backups --backup-id` *`backup_ID`*. Les résultats de la commande montrent l'ARN de la sauvegarde. Vous pouvez également exécuter `describe-backups --server-name` *`server_name`* pour afficher des informations sur toutes les sauvegardes d'un AWS OpsWorks for Chef Automate serveur spécifique.

2. Exécutez la commande `untag-resource` avec l'ARN que vous avez renvoyé à l'étape 1. Spécifiez uniquement les balises que vous souhaitez supprimer.

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

   Dans cet exemple, la commande `untag-resource` supprime uniquement la balise dont la clé est `Stage` et dont la valeur est `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. Pour vérifier que les balises ont été supprimées avec succès, suivez les étapes décrites dans Liste des balises de ressource de la présente rubrique.

consultez aussi

- Créez un serveur Chef Automate à l'aide du AWS CLI
- Sauvegarder un AWS OpsWorks for Chef Automate serveur
- Stratégies de balisage AWS
- Contrôle de l'accès aux AWS ressources à l'aide des balises de ressources dans le guide de AWS Identity and Access Management l'utilisateur
- Utilisation des balises de répartition des coûts dans le Guide de l'utilisateur AWS Billing and Cost Management .
- CreateBackup dans la Référence d'API AWS OpsWorks CM
- CreateServer dans la Référence d'API AWS OpsWorks CM
- TagResource dans la Référence d'API AWS OpsWorks CM
- ListTagsForResource dans la Référence d'API AWS OpsWorks CM
- UntagResource dans la Référence d'API AWS OpsWorks CM

Sauvegarder et restaurer un AWS OpsWorks for Chef Automate serveur

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section explique comment sauvegarder et restaurer un AWS OpsWorks for Chef Automate serveur, puis supprimer des sauvegardes.

Rubriques

- Sauvegarder un AWS OpsWorks for Chef Automate serveur
- Restaurer un AWS OpsWorks for Chef Automate serveur à partir d'une sauvegarde

Sauvegarder un AWS OpsWorks for Chef Automate serveur

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez définir une sauvegarde de AWS OpsWorks for Chef Automate serveur récurrente, qu'elle soit quotidienne ou hebdomadaire, puis demander au service de stocker les sauvegardes dans Amazon Simple Storage Service (Amazon S3) en votre nom. Sinon, vous pouvez faire effectuer des sauvegardes manuelles à la demande.

Les sauvegardes étant stockées dans Amazon S3, des frais supplémentaires s'appliquent. Vous pouvez définir une période de rétention des sauvegardes de 30 générations maximum. Vous pouvez soumettre une demande de service pour faire modifier cette limite en utilisant les canaux d'

AWS assistance. Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Vous pouvez ajouter des balises aux sauvegardes d'un AWS OpsWorks for Chef Automate serveur. Si vous avez ajouté des balises à un serveur AWS OpsWorks for Chef Automate , les sauvegardes automatisées du serveur héritent de ces balises. Pour de plus amples informations sur l'ajout et la gestion de balises sur les sauvegardes, veuillez consulter Utilisation des balises sur les AWS OpsWorks for Chef Automate ressources dans ce guide.

Rubriques

- Sauvegardes automatisées
- Sauvegardes manuelles
- Supprimer des sauvegardes

Sauvegardes automatisées

Lorsque vous configurez votre AWS OpsWorks for Chef Automate serveur, vous choisissez des sauvegardes automatisées ou manuelles. AWS OpsWorks for Chef Automate lance les sauvegardes automatisées pendant l'heure et le jour que vous avez choisis dans la section Sauvegarde automatique de la page Configurer les paramètres avancés du programme d'installation. Une fois que votre serveur est en ligne, vous pouvez modifier les paramètres de sauvegarde en effectuant les étapes suivantes à partir de la vignette du serveur sur la page d'accueil des serveurs Chef Automate ou sur la page des propriétés du serveur.

Pour modifier les paramètres de sauvegarde automatisée

1. Dans le menu Actions de la vignette du serveur sur la page d'accueil Serveurs Chef, choisissez Modifier les paramètres
2. Pour désactiver les sauvegardes automatisées, choisissez Non pour l'option Activer les sauvegardes automatiques. Enregistrez'vos modifications ; vous n'avez pas besoin de passer à l'étape suivante.
3. Dans la section Sauvegarde automatique, modifiez la fréquence, l'heure de début ou les générations à conserver. Enregistrez vos modifications.

Sauvegardes manuelles

Vous pouvez démarrer une sauvegarde manuelle à tout moment dans le ou AWS Management Console en exécutant la commande AWS CLI create-backup. Les sauvegardes manuelles ne sont pas incluses dans les 30 générations maximum de sauvegardes automatisées stockées ; un maximum de 10 sauvegardes manuelles sont stockées et doivent être supprimées manuellement d'Amazon S3.

Vous pouvez ajouter des balises lorsque vous créez une sauvegarde manuelle d'un AWS OpsWorks for Chef Automate serveur. Pour de plus amples informations sur l'ajout de balises lors de la création d'une sauvegarde manuelle, reportez-vous à Ajouter des balises à une nouvelle sauvegarde (interface de ligne de commande).

Pour effectuer une sauvegarde manuelle dans AWS Management Console

1. Sur la page Serveurs Chef Automate, sélectionnez le serveur à sauvegarder.
2. Sur la page de propriétés du serveur, sélectionnez Sauvegardes dans le panneau de navigation de gauche.
3. Choisissez Créer une sauvegarde.
4. La sauvegarde manuelle est terminée lorsque la page affiche une coche verte dans la colonne Statut de la sauvegarde.

Pour effectuer une sauvegarde manuelle dans AWS CLI

- Pour démarrer une sauvegarde manuelle, exécutez la AWS CLI commande suivante.

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

Supprimer des sauvegardes

Si vous supprimez une sauvegarde, celle-ci est définitivement éliminée du compartiment S3 où les sauvegardes sont stockées.

Pour supprimer une sauvegarde dans AWS Management Console

1. Sur la page Serveurs Chef Automate, sélectionnez le serveur à sauvegarder.

2. Sur la page de propriétés du serveur, sélectionnez Sauvegardes dans le panneau de navigation de gauche.
3. Sélectionnez la sauvegarde que vous voulez supprimer et choisissez Supprimer une sauvegarde. Vous ne pouvez sélectionner qu'une seule sauvegarde à la fois.
4. Lorsque vous êtes invité à confirmer la suppression, cochez la case Delete the backup, which is stored in an S3 bucket (Supprimer la sauvegarde stockée dans un compartiment S3), puis sélectionnez Oui, supprimer.

Pour supprimer une sauvegarde dans AWS CLI

- Pour supprimer une sauvegarde, exécutez la AWS CLI commande suivante en la `--backup-id` remplaçant par l'ID de la sauvegarde que vous souhaitez supprimer. IDs Les sauvegardes sont au format*ServerName-yyyyMMddHHmmssSSS*. Par exemple, **`test-chef-server-20171218132604388`**.

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

Restaurer un AWS OpsWorks for Chef Automate serveur à partir d'une sauvegarde

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Après avoir parcouru les sauvegardes disponibles, vous pouvez choisir un moment à partir duquel vous souhaitez restaurer votre AWS OpsWorks for Chef Automate serveur. Les sauvegardes de serveur contiennent uniquement les données persistantes de logiciel de gestion de configuration (livre de recettes, nœuds enregistrés, etc.). L'exécution d'une restauration sur place d'un serveur (c'est-à-dire la restauration AWS OpsWorks for Chef Automate du serveur existant sur une nouvelle EC2 instance) permet de réenregistrer les nœuds enregistrés au moment de la sauvegarde que vous

utilisez pour restaurer le serveur, et de transférer le trafic vers la nouvelle instance si la restauration est réussie et si l'état du serveur restauré AWS OpsWorks for Chef Automate l'est. `Healthy` La restauration d'un serveur AWS OpsWorks for Chef Automate nouvellement créé ne conserve pas les connexions de nœud. La restauration d'un serveur ne permet pas de mettre à jour les versions mineures du logiciel Chef ; elle applique les mêmes versions de Chef et les données de gestion de configuration qui sont disponibles dans la sauvegarde que vous choisissez.

La restauration d'un serveur prend généralement plus de temps que la création d'un nouveau serveur ; le temps dépend de la taille de la sauvegarde que vous choisissez. Une fois la restauration terminée, l'ancienne EC2 instance reste à `Stopped` l'état `Running` or, mais uniquement temporairement. À terme, elle est résiliée.

Dans cette version, vous pouvez utiliser le AWS CLI pour restaurer un serveur Chef dans AWS OpsWorks for Chef Automate.

> **Note**
>
> Vous pouvez également exécuter la commande restore-server pour modifier le type d'instance actuelle, ou pour restaurer ou définir votre clé SSH si elles est perdue ou divulguée.

Pour restaurer un serveur à partir d'une sauvegarde

1. Dans le AWS CLI, exécutez la commande suivante pour renvoyer une liste des sauvegardes disponibles et leurs IDs. Notez l'ID de la sauvegarde que vous souhaitez utiliser. IDs Les sauvegardes sont au format*mySerName-yyyyMMddHHmmssSSS*.

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

2. Exécutez la commande suivante.

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

   Voici un exemple.

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

3. Attendez que la restauration soit terminée.

Maintenance du système dans AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

La maintenance obligatoire du système garantit que les dernières versions mineures de Chef Server et Chef Automate Server, y compris les mises à jour de sécurité, s'exécutent toujours sur un AWS OpsWorks for Chef Automate serveur. La maintenance du système doit avoir lieu au moins une fois par semaine. En utilisant le AWS CLI, vous pouvez configurer la maintenance automatique quotidienne, si vous le souhaitez. Vous pouvez également utiliser le AWS CLI pour effectuer la maintenance du système à la demande, en plus de la maintenance planifiée du système.

Lorsque de nouvelles versions mineures du logiciel Chef deviennent disponibles, la maintenance système est conçue pour mettre à jour automatiquement la version mineure de Chef Automate et de Chef Server sur le serveur dès que celle-ci réussit les tests AWS. AWS effectue des tests approfondis pour vérifier que les mises à niveau de Chef sont prêtes pour la production et ne perturbent pas les environnements clients existants. Il peut donc y avoir des retards entre les versions du logiciel Chef et leur disponibilité en tant qu'application par rapport aux serveurs existants OpsWorks pour les serveurs Chef Automate. Pour mettre à jour les versions mineures du logiciel Chef à la demande, consultez Démarrage de la maintenance du système à la demande dans cette rubrique.

La maintenance du système lance une nouvelle instance à partir d'une sauvegarde effectuée dans le cadre du processus de maintenance, ce qui permet de réduire les risques liés à la dégradation ou à l'altération des EC2 instances Amazon soumises à une maintenance périodique.

> **Important**
>
> Le système de maintenance supprime les éventuels fichiers ou configurations personnalisés que vous avez ajoutés au serveur AWS OpsWorks for Chef Automate . Pour plus d'informations sur la résolution d'un problème de perte de configuration ou de fichier, consultez Restauration des configurations personnalisées et des fichiers après la maintenance dans cette rubrique.

Rubriques

- S'assurer que les nœuds font confiance à l'autorité AWS OpsWorks de certification
- Configuration de la maintenance du système
- Démarrage de la maintenance du système à la demande
- Restauration des configurations personnalisées et des fichiers après la maintenance

S'assurer que les nœuds font confiance à l'autorité AWS OpsWorks de certification

> **Note**
>
> Les étapes décrites dans cette section ne sont pas obligatoires si vous utilisez un domaine et un certificat personnalisés avec votre AWS OpsWorks for Chef Automate serveur.

Les nœuds que vous gérez avec un AWS OpsWorks for Chef Automate serveur doivent s'authentifier auprès du serveur à l'aide de certificats. Pendant la maintenance du système, AWS OpsWorks remplace l'instance du serveur et régénère les nouveaux certificats par le biais de l'autorité de AWS OpsWorks certification (CA). Pour rétablir automatiquement la communication avec vos nœuds gérés une fois la maintenance terminée, les nœuds doivent faire confiance à l' AWS OpsWorks autorité de certification fournie avec le kit de démarrage et hébergée dans les régions prises en charge par AWS OpsWorks for Chef Automate. Lorsque vous utilisez l' AWS OpsWorks autorité de certification pour établir la confiance entre les nœuds et le serveur, les nœuds se reconnectent à la nouvelle instance de serveur après la maintenance. Si vous ajoutez des EC2 nœuds à l'aide du EC2 `userdata` script décrit dansAjouter des nœuds automatiquement AWS OpsWorks for Chef Automate, les nœuds sont déjà configurés pour faire confiance à l' AWS OpsWorks autorité de certification.

- Pour les nœuds basés sur Linux, l'emplacement du compartiment S3 de l'autorité de certification est `https://opsworks-cm-${REGION}-prod-default-assets.s3.amazonaws.com/misc/opsworks-cm-ca-2020-root.pem`. L'autorité AWS OpsWorks de certification approuvée doit être enregistrée dans le chemin`/etc/chef/opsworks-cm-ca-2020-root.pem`.
- Pour les nœuds basés sur Windows, l'emplacement du compartiment S3 de l'autorité de certification est `https://opsworks-cm-$env:AWS_REGION-prod-default-assets.s3.amazonaws.com/misc/opsworks-cm-ca-2020-root.pem`. L' AWS OpsWorks autorité de certification doit être stockée dans le dossier racine du chef ; par exemple, `C:\chef\opsworks-cm-ca-2020-root.pem`

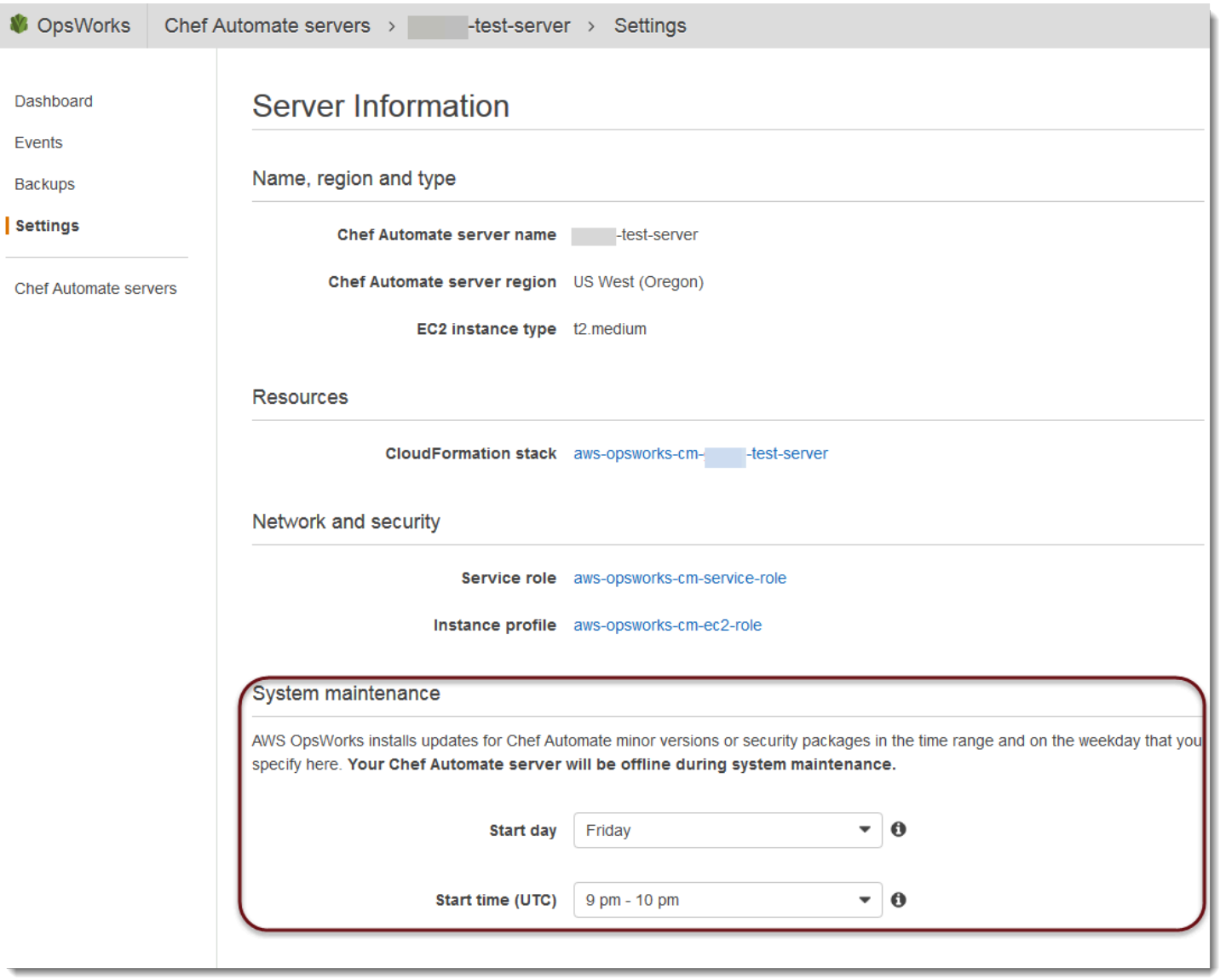
Dans les deux emplacements, la variable de région est résolue de l'une des manières suivantes :

- `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`

Configuration de la maintenance du système

Lorsque vous créez un nouveau AWS OpsWorks for Chef Automate serveur, vous pouvez configurer un jour et une heure de la semaine, en temps universel coordonné (UTC), pour le début de la maintenance du système. La maintenance commence au cours de l'heure que vous spécifiez. Etant donné que le serveur doit être hors ligne lors de la maintenance du système, choisissez une heure où le serveur est peu sollicité pendant les heures normales de bureau. L'état du serveur est `UNDER_MAINTENANCE` lorsque la maintenance est en cours.

Vous pouvez également modifier les paramètres de maintenance du système sur un AWS OpsWorks for Chef Automate serveur existant, en modifiant les paramètres dans la zone Maintenance du système de la page Paramètres de votre serveur, comme indiqué dans la capture d'écran suivante.



Dans la section System maintenance (Maintenance du système), définissez le jour et l'heure de début de la maintenance du système.

Configuration de la maintenance du système à l'aide du AWS CLI

Vous pouvez également configurer le démarrage automatique du système de maintenance à l'aide de l' AWS CLI. Vous AWS CLI permet de configurer la maintenance automatique quotidienne, si vous le souhaitez, en omettant le préfixe à trois caractères du jour de la semaine.

Dans une commande `create-server`, ajoutez le paramètre `--preferred-maintenance-window` à votre commande, après avoir indiqué les exigences relatives à la création de l'instance serveur (par exemple, le type d'instance, l'ARN du profil d'instance et l'ARN du rôle de service). Dans l'exemple `create-server` suivant, la valeur définie pour `--preferred-maintenance-window`

est `Mon:08:00`, ce qui signifie que vous avez défini le début de la maintenance chaque lundi matin à 8 h 00. UTC.

```
aws opsworks-cm create-server --engine "Chef" --engine-model "Single" --
engine-version "12" --server-name "automate-06" --instance-profile-arn
 "arn:aws:iam::1019881987024:instance-profile/aws-opsworks-cm-ec2-role"
 --instance-type "t2.medium" --key-pair "amazon-test" --service-role-arn
 "arn:aws:iam::044726508045:role/aws-opsworks-cm-service-role" --preferred-maintenance-
window "Mon:08:00"
```

Dans une commande `update-server`, vous pouvez mettre à jour la valeur `--preferred-maintenance-window` seulement, si vous le souhaitez. Dans l'exemple suivant, la fenêtre de maintenance est définie à vendredi soir à 18 h 15. UTC.

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

Pour modifier l'heure de début de la fenêtre de maintenance à 18 h 15 chaque jour, n'indiquez pas le préfixe à trois caractères du jour de la semaine, comme illustré dans l'exemple suivant.

```
aws opsworks-cm update-server --server-name "shiny-kitchen" --preferred-maintenance-
window "18:15"
```

Pour plus d'informations sur la définition de la fenêtre de maintenance système préférée à l'aide du AWS CLI, voir create-server et update-server.

Démarrage de la maintenance du système à la demande

Pour démarrer la maintenance du système à la demande, en dehors de votre maintenance automatique hebdomadaire ou quotidienne configurée, exécutez la AWS CLI commande suivante. Vous ne pouvez pas démarrer la maintenance à la demande dans AWS Management Console.

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

Pour plus d'informations sur cette commande, consultez start-maintenance.

Restauration des configurations personnalisées et des fichiers après la maintenance

La maintenance du système peut supprimer ou modifier les fichiers personnalisés ou les configurations que vous avez ajoutés à votre AWS OpsWorks for Chef Automate serveur.

Si, après une opération de maintenance, il manque à votre serveur Chef des fichiers ou des paramètres que vous avez ajoutés à l'aide `RunCommand` de SSH, vous pouvez utiliser une Amazon Machine Image (AMI) pour lancer une nouvelle EC2 instance Amazon. AMIs sont disponibles et sont créés à partir de la configuration préalable à la maintenance d'un serveur.

L'état de la nouvelle instance est celui du serveur Chef avant la maintenance. La nouvelle instance doit inclure vos fichiers et paramètres manquants.

> **Important**
>
> Vous ne pouvez pas utiliser la nouvelle instance pour restaurer votre serveur ; l'instance ne peut pas être exécutée en tant que serveur Chef. Vous pouvez utiliser l'instance uniquement pour récupérer vos fichiers et les paramètres de configuration.

Pour lancer une EC2 instance depuis une AMI, dans la EC2 console Amazon, ouvrez l'assistant de lancement, choisissez My AMIs, puis choisissez l'AMI qui porte le nom de votre serveur. Suivez les étapes de EC2 l'assistant Amazon comme vous le feriez pour tout autre lancement d'instance.

Analyses de conformité dans AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les analyses de conformité vous permettent de suivre la conformité des nœuds gérés de votre infrastructure en fonction de stratégies prédéfinies, également appelées règles. Grâce aux vues de conformité, vous pouvez auditer régulièrement vos applications pour repérer des vulnérabilités et des

configurations non conformes. Chef propose plus de 100 profils de conformité prédéfinis (c'est-à-dire des ensembles de règles qui s'appliquent à des configurations de nœud spécifiques) que vous pouvez utiliser dans vos analyses de conformité. Vous pouvez également utiliser le InSpec langage Chef pour créer vos propres profils personnalisés.

Si votre serveur n'exécute pas encore Chef Automate 2.0, vous pouvez configurer Chef Compliance manuellement, en installant le livre de recettes Audit.

> **Note**
>
> La version minimale prise en charge du logiciel d'agent client Chef Infra (`chef-client`) sur les nœuds associés à un AWS OpsWorks for Chef Automate serveur est de 13. x. Nous vous recommandons d'utiliser la `chef-client` version stable la plus récente, ou au moins 14.10.9.

Rubriques

- Conformité dans Chef Automate 2.0
- Conformité dans Chef Automate 1.x
- Mises à jour de la conformité
- Profils de conformité personnalisés et de la communauté
- consultez aussi

Conformité dans Chef Automate 2.0

Si votre AWS OpsWorks for Chef Automate serveur exécute Chef Automate 2.0, configurez Chef Compliance en utilisant les procédures décrites dans cette section.

Exécution de tâches d'analyse de conformité avec Chef Automate 2.0

Chef Automate 2.0 inclut la fonctionnalité InSpec d'analyse de conformité de Chef qui nécessitait auparavant une configuration manuelle et une configuration de livre de recettes. Vous pouvez exécuter des tâches de numérisation sur un AWS OpsWorks for Chef Automate serveur qui exécute Chef Automate 2.0. Les tâches peuvent être exécutées immédiatement (une fois), planifiées à une date ultérieure ou planifiées pour s'exécuter à intervalles définis, par exemple quotidiennement ou toutes les deux heures. Les résultats d'une tâche d'analyse sont envoyés dans un rapport de conformité. Vous pouvez afficher les résultats de l'analyse de conformité et effectuer des actions en

conséquence dans le tableau de bord Chef Automate. Pour ouvrir l'onglet Compliance (Conformité) et afficher les rapports, sous l'onglet Scan Jobs (Tâches d'analyse) du tableau de bord Chef Automate, choisissez Report (Rapport) à droite de la ligne d'un nœud géré.

Pour exécuter des tâches d'analyse sur des nœuds gérés, vous devez disposer des éléments suivants.

- Au moins un profil de conformité installé dans votre espace de noms.
- Au moins un nœud cible, soit ajouté manuellement, soit une EC2 instance ajoutée automatiquement.

Dans AWS OpsWorks for Chef Automate, les tâches de numérisation sont prises en charge sur les cibles suivantes.

- Nœuds ajoutés manuellement
- Instances `aws-ec2`
- Régions AWS

Pour obtenir des instructions détaillées sur l'exécution des tâches d'analyse, consultez Tâches d'analyse Chef Automate dans la documentation Chef.

(Facultatif, Chef Automate 2.0) Configuration de la conformité avec le livre de recettes Audit

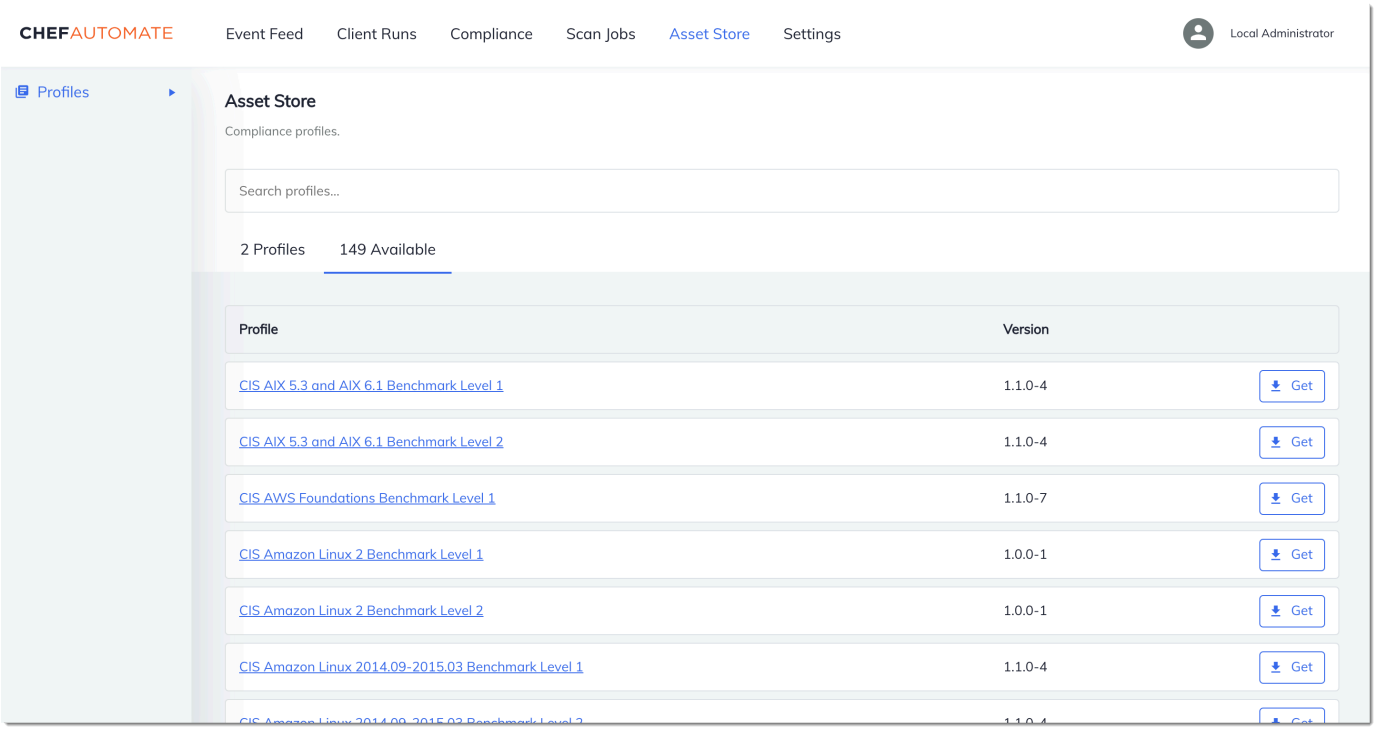
Vous pouvez configurer la conformité sur n'importe quel AWS OpsWorks for Chef Automate serveur. Après avoir lancé un serveur AWS OpsWorks for Chef Automate , vous pouvez installer les profils à partir du tableau de bord Chef Automate ou ajouter les profils souhaités aux attributs du livre de recettes Audit dans le fichier de stratégie `Policyfile.rb`. Un fichier `Policyfile.rb` prérempli est inclus dans le kit de démarrage.

Après avoir modifié le fichier `Policyfile.rb` avec les profils à utiliser comme attributs du livre de recettes Audit, exécutez les commandes `chef push` pour charger le livre de recettes Audit et les autres livres de recettes spécifiés dans `Policyfile.rb` sur votre serveur Chef Automate. L'installation du livre de recettes Audit permet également d'installer la gemme de Chef InSpec, un framework de test et d'audit open source produit par Chef. Pour Chef Automate 2.0, choisissez la version 7.1.0 ou ultérieure du livre de recettes Audit. La InSpec gemme doit être de version 2.2.102 ou ultérieure.

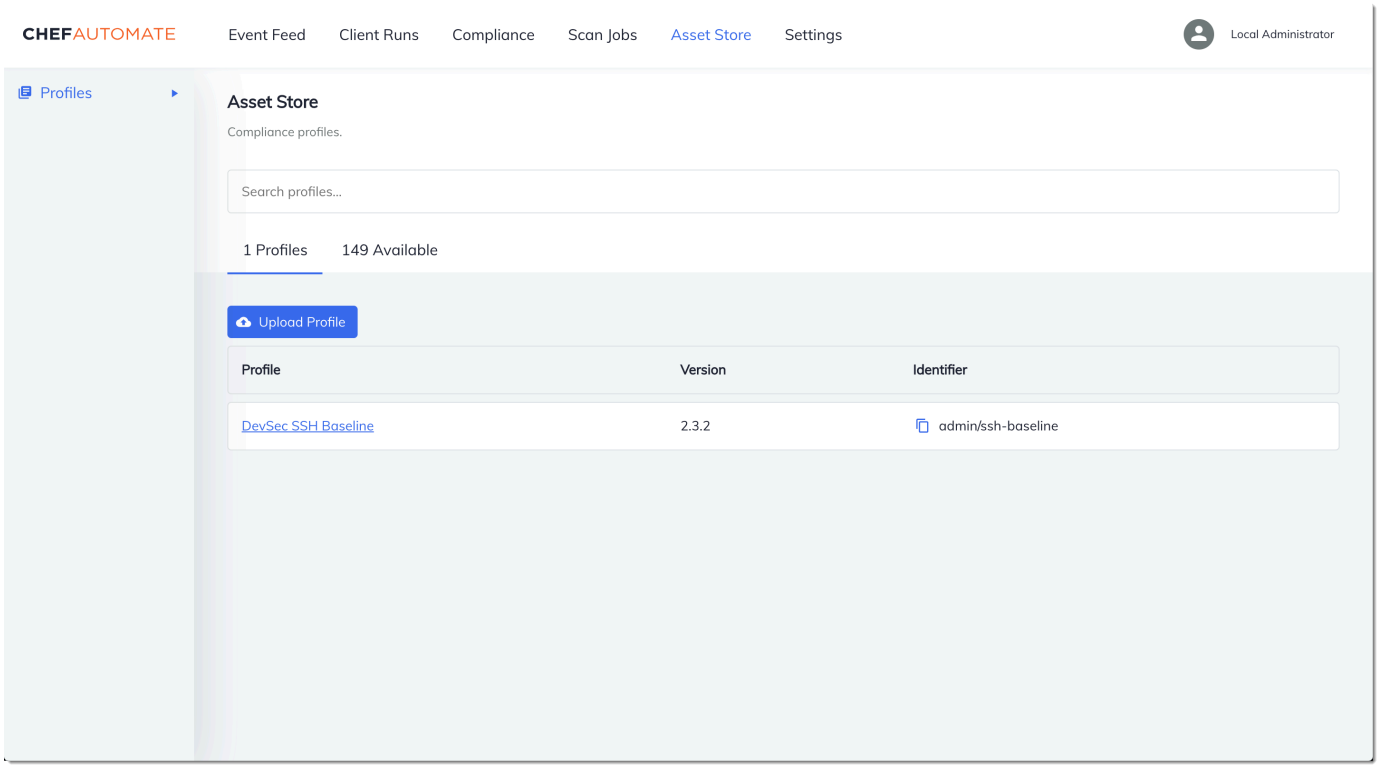
Les instructions de cette section vous montrent comment implémenter le livre de recettes `opsworks-audit`. Le livre de recettes Audit télécharge les profils spécifiés depuis le serveur Chef Automate, évalue les nœuds par rapport au profil de base DevSec SSH et rapporte le résultat des analyses de conformité à chaque exécution. `chef-client`

Pour installer les profils de conformité

1. Si vous ne l'avez pas déjà fait, connectez-vous au tableau de bord web Chef Automate. Utilisez les informations d'identification que vous avez reçues lorsque vous avez téléchargé le Starter Kit lors de la création de votre serveur AWS OpsWorks for Chef Automate .
2. Dans le tableau de bord Chef Automate, choisissez l'onglet Asset Store (Magasin de ressources).



3. Choisissez l'onglet Available (Disponibles) pour afficher les profils prédéfinis.
4. Parcourez la liste des profils. Choisissez un profil correspondant au système d'exploitation et à la configuration d'au moins un de vos nœuds gérés. Pour afficher des détails sur le profil, dont une description des violations ciblées et le code des règles sous-jacentes, choisissez **>** à droite de l'entrée du profil. Vous pouvez choisir plusieurs profils. Si vous configurez l'exemple dans le kit de démarrage, choisissez DevSec SSH Baseline.



5. Pour installer les profils sélectionnés sur votre serveur Chef Automate, choisissez Get (Obtenir).
6. Une fois que vous avez installé les profils, ils s'affichent dans l'onglet Profiles (Profils) du tableau de bord Chef Automate.

Pour installer des livres de recettes avec **`Policyfile.rb`**

1. Affichez le fichier `Policyfile.rb` de votre Starter Kit pour voir que les attributs du livre de recettes Audit spécifient le profil `ssh-baseline` dans `['profiles']`.

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

2. Téléchargez et installez les livres de recettes définis dans `Policyfile.rb`.

```
chef install
```

Tous les livres de recettes sont versionnés dans le fichier `metadata.rb` du livre de recettes. Chaque fois que vous modifiez un livre de recettes, vous devez modifier la version du livre de recettes se trouvant dans son fichier `metadata.rb`.

3. Transférez la stratégie `opsworks-demo`, définie dans `Policyfile.rb`, à votre serveur.

```
chef push opsworks-demo
```

4. Vérifiez l'installation de votre stratégie. Exécutez la commande suivante.

```
chef show-policy
```

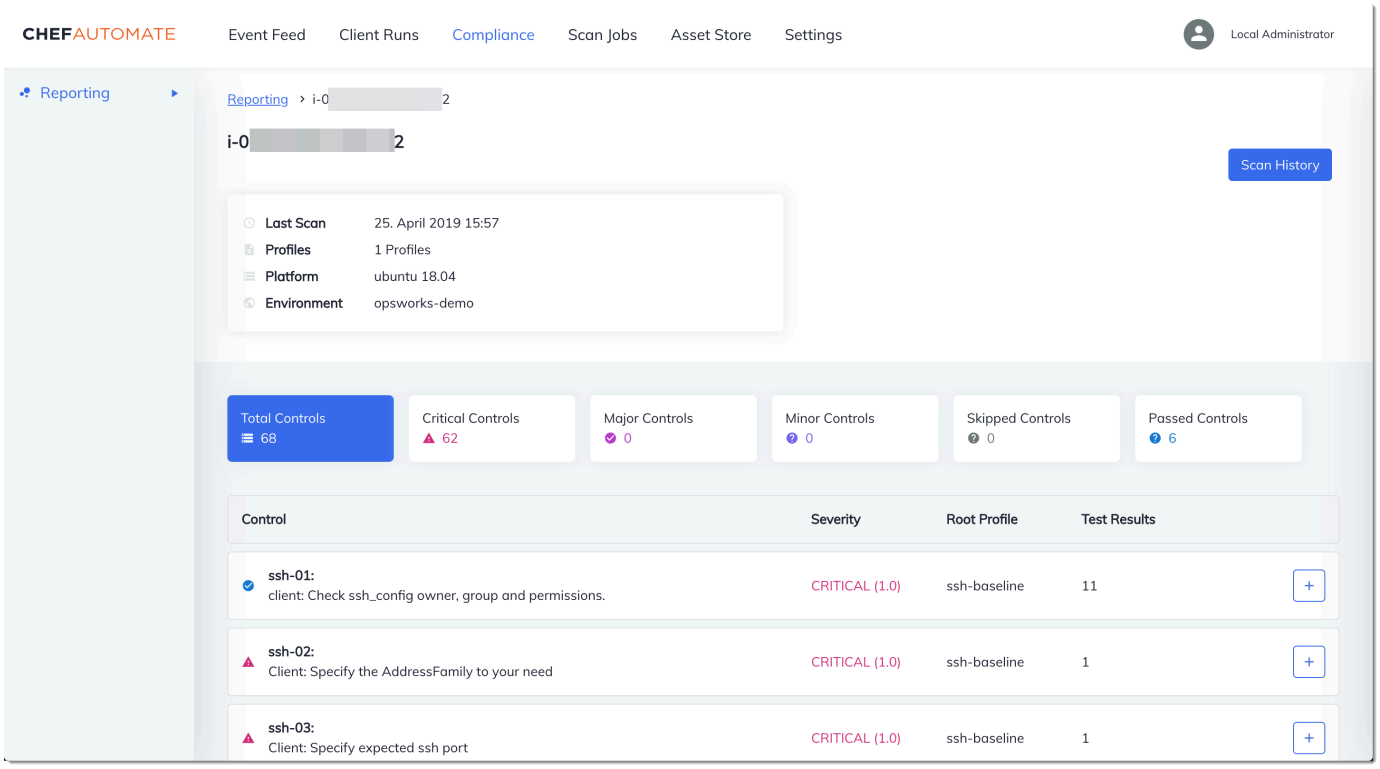
Les résultats doivent ressembler à ce qui suit :

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

5. Ajoutez des nœuds à votre serveur pour les gérer, si vous ne l'avez pas déjà fait. Pour connecter votre premier nœud au AWS OpsWorks for Chef Automate serveur, utilisez le `userdata.sh` script inclus dans ce kit de démarrage. Il utilise l' AWS OpsWorks `AssociateNode`API pour connecter un nœud à votre serveur.

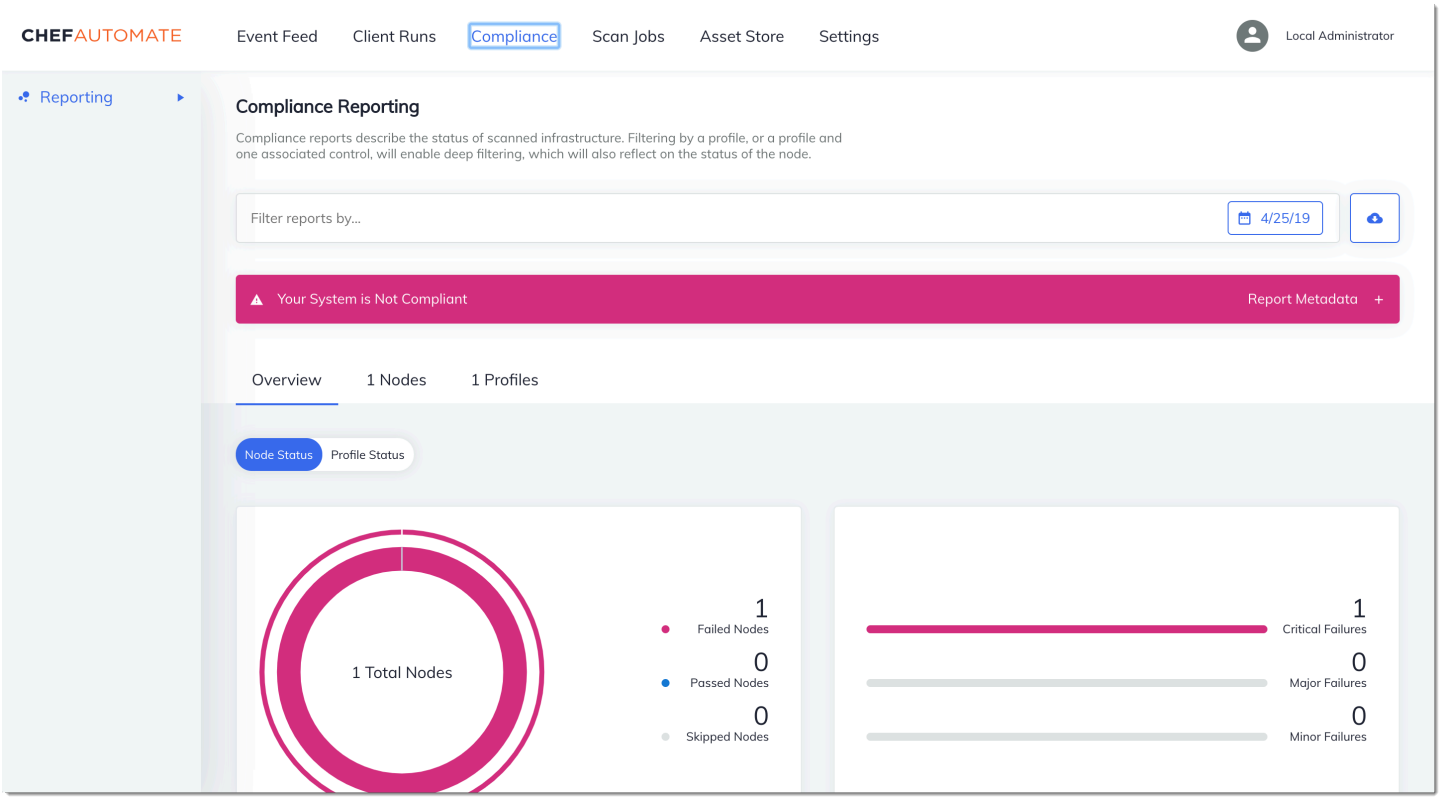
   Vous pouvez automatiser l'association des nœuds en suivant les étapes décrites dans Ajouter des nœuds automatiquement AWS OpsWorks for Chef Automate ou ajouter des nœuds l'un après l'autre en suivant les étapes indiquées dans Ajouter des nœuds individuellement.

6. Une fois que vous avez mis à jour la liste d'exécution de vos nœuds, l'agent `chef-client` exécute les recettes spécifiées lors de son exécution suivante. Par défaut, cela se produit toutes les 1 800 secondes (30 minutes). Après l'exécution, vous pouvez afficher les résultats de conformité et effectuer des actions en conséquence à partir de l'onglet Compliance (Conformité) du tableau de bord Chef Automate.

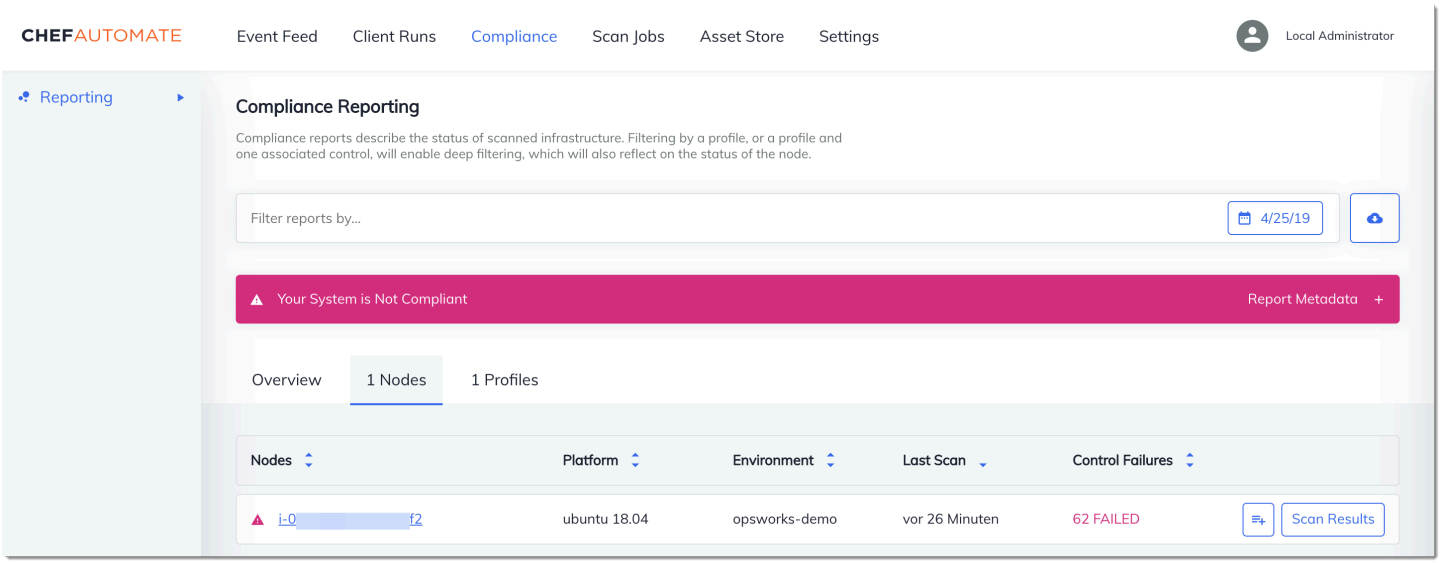


Exécution d'une analyse de conformité

Vous devriez voir les résultats de l'analyse de conformité dans le tableau de bord Chef Automate peu après la première exécution de l'agent, qui se produit une fois que vous avez configuré les listes d'exécution des nœuds.



Dans le tableau de bord Chef Automate, choisissez l'onglet Compliance (Conformité). Dans le panneau de navigation de gauche, choisissez Reporting (Génération de rapports). Choisissez l'onglet Profiles (Profils), sélectionnez Scan Results (Résultats de l'analyse), puis choisissez un nœud présentant des échecs d'analyse pour en savoir plus sur les règles ayant entraîné l'échec du nœud.



Généralement, les résultats d'analyse ne sont pas conformes, car les nouveaux nœuds ne satisfont pas encore à toutes les règles du profil de base DevSec SSH. Le DevSec Hardening Framework, un

projet communautaire, propose des livres de recettes pour résoudre les problèmes qui enfreignent les règles du profil de base DevSec SSH.

(Facultatif) Résolution des résultats non conformes

Le kit de démarrage comprend un livre de recettes open source que vous pouvez exécuter pour corriger les résultats non conformes résultant d'exécutions effectuées avec le profil de base DevSec SSH. `ssh-hardening`

> **Note**
>
> Le `ssh-hardening` livre de recettes apporte des modifications à vos nœuds pour les mettre en conformité avec les règles de base DevSec SSH. Avant d'exécuter ce livre de recettes sur un nœud de production, consultez les détails du profil de base DevSec SSH dans la console Chef Automate pour comprendre les violations des règles ciblées par le livre de recettes. Consultez les informations sur le livre de recettes open source `ssh-hardening` avant de l'exécuter sur un nœud de production.

Pour exécuter le livre de recettes **`ssh-hardening`**

1. Dans un éditeur de texte, ajoutez le livre de recettes `ssh-hardening` à la liste d'exécution de `Policyfile.rb`. La liste d'exécution de `Policyfile.rb` doit correspondre à ce qui suit.

   ```
   run_list  'chef-client', 'opsworks-webserver', 'audit', 'ssh-hardening'
   ```

2. Mettez à jour le fichier `Policyfile.rb` et transmettez-le à votre serveur AWS OpsWorks for Chef Automate .

   ```
   chef update Policyfile.rb
      chef push opsworks-demo
   ```

3. Les nœuds qui sont associés à la stratégie `opsworks-demo` mettent automatiquement à jour la liste d'exécution et appliquent le livre de recettes `ssh-hardening` lors de l'exécution suivante de `chef-client`.

   Étant donné que vous utilisez le livre de recettes `chef-client`, votre nœud effectue des vérifications à intervalles réguliers (par défaut, toutes les 30 minutes). Lors du prochain enregistrement, le `ssh-hardening` livre de recettes s'exécute et contribue à améliorer la sécurité des nœuds afin de respecter les règles du profil DevSec SSH Baseline.

4. Une fois l'exécution initiale du livre de recettes `ssh-hardening` terminée, attendez 30 minutes afin qu'une analyse de conformité s'exécute à nouveau. Affichez les résultats dans le tableau de bord Chef Automate. Les résultats non conformes survenus lors de l'exécution initiale de l'analyse DevSec SSH Baseline doivent être résolus.

Conformité dans Chef Automate 1.x

Si votre AWS OpsWorks for Chef Automate serveur exécute Chef Automate 1. x, configurez Chef Compliance en utilisant les procédures décrites dans cette section.

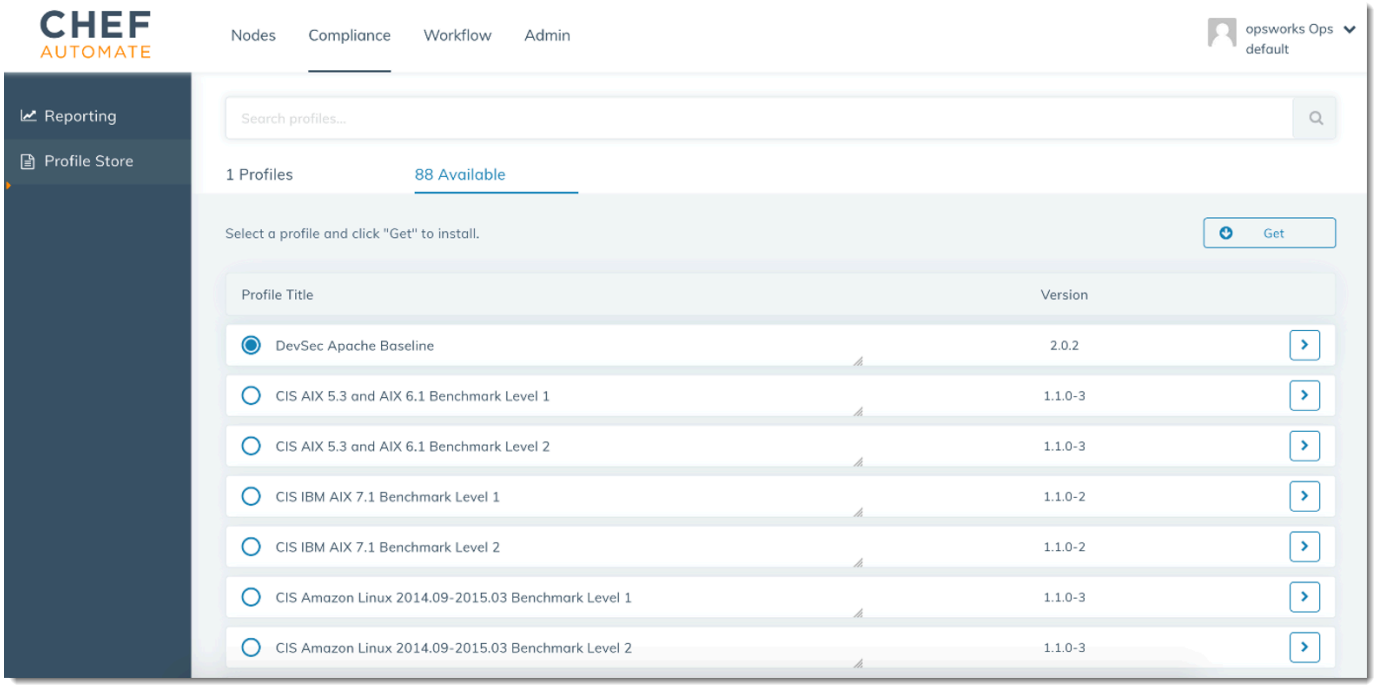
(Facultatif, Chef Automate 1.x) Configuration de Chef Compliance

Vous pouvez configurer Chef Compliance sur n'importe quel AWS OpsWorks for Chef Automate serveur. Après avoir lancé un serveur AWS OpsWorks for Chef Automate , choisissez les profils que vous souhaitez exécuter parmi ceux du tableau de bord Chef Automate. Après avoir installé les profils, exécutez les commandes `berks` pour charger le livre de recettes Audit sur votre serveur Chef Automate. L'installation du livre de recettes Audit permet également d'installer le gem for InSpec, un framework de test open source produit par Chef qui vous permet d'intégrer des tests automatisés à n'importe quelle étape de votre pipeline de déploiement. Pour Chef Automate 1.x, choisissez la version 5.0.1 ou ultérieure du livre de recettes Audit. La InSpec gem doit être de version 1.24.0 ou ultérieure.

Le kit de AWS OpsWorks for Chef Automate démarrage comprend un livre de recettes emballé qui télécharge et installe la version du livre de recettes Chef's Audit qui vous convient. `opsworks-audit` Le `opsworks-audit` livre de recettes demande également à l'`chef-client`agent d'évaluer les nœuds par rapport au profil de base DevSec SSH que vous installez à partir de la console de conformité de Chef plus loin dans cette rubrique. Vous pouvez configurer la conformité en utilisant le livre de recettes de votre choix. Les instructions de cette section vous montrent comment implémenter le livre de recettes `opsworks-audit`.

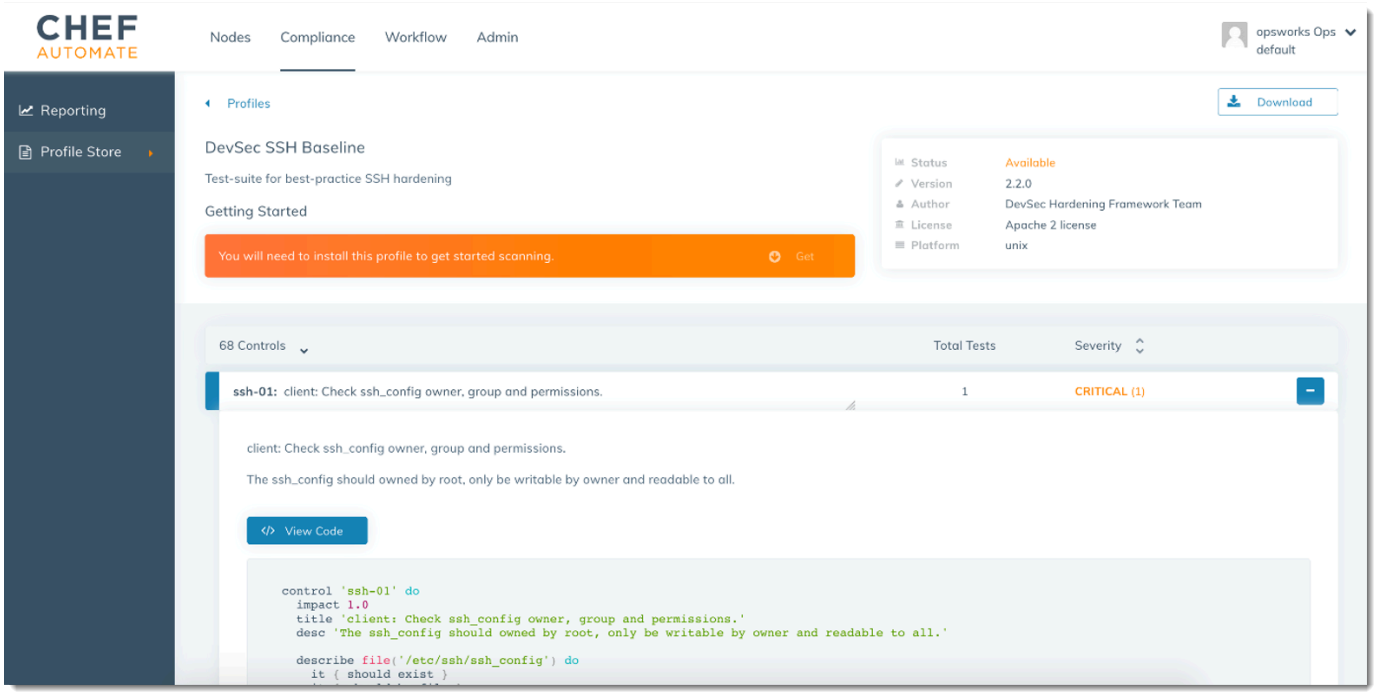
Pour installer les profils de conformité

1. Si vous ne l'avez pas déjà fait, connectez-vous au tableau de bord web Chef Automate. Utilisez les informations d'identification que vous avez reçues lorsque vous avez téléchargé le kit de démarrage lors de la création de votre AWS OpsWorks for Chef Automate serveur.
2. Dans le tableau de bord Chef Automate, choisissez l'onglet Compliance (Conformité).



3. Dans la barre de navigation de gauche, choisissez Profile Store (Stockage de profils), puis ouvrez l'onglet Disponible pour afficher les profils prédéfinis disponibles.

4. Parcourez la liste des profils. Choisissez un profil correspondant au système d'exploitation et à la configuration d'au moins un de vos nœuds gérés. Pour afficher des détails sur le profil, dont une description des violations ciblées et le code des règles sous-jacentes, choisissez **>** à droite de l'entrée du profil. Vous pouvez choisir plusieurs profils.



5. Pour installer les profils sélectionnés sur votre serveur Chef Automate, choisissez Get (Obtenir).
6. Lorsque le téléchargement est terminé, passez à la procédure suivante.

Pour installer et configurer le livre de recettes **`opsworks-audit`**

1. Cette étape est facultative, mais elle permet de gagner du temps à l'étape 6, lorsque vous ajoutez des recettes aux listes d'exécution des nœuds. Modifiez le fichier `roles/opsworks-example-role.rb` qui est inclus dans le kit de démarrage que vous avez téléchargé lors de la création de votre serveur AWS OpsWorks for Chef Automate . Ajoutez les lignes suivantes. La dernière ligne est mise en commentaire, car l'ajout du livre de recettes `ssh-hardening` et des recettes correspondantes pour résoudre les nœuds non conformes après l'exécution de votre analyse de conformité est facultatif.

   ```
   run_list(
        "recipe[chef-client]",
        "recipe[apache2]",
        "recipe[opsworks-audit]"
        # "recipe[ssh-hardening]"
          )
   ```

2. Utilisez un éditeur de texte pour spécifier les livres de recettes souhaités dans votre Berksfile. Un exemple de fichier Berksfile est fourni dans le kit de démarrage. Dans cet exemple, nous installons le livre de recettes du client Chef Infra (`chef-client`), ainsi que les livres de recettes `apache2` et `opsworks-audit`. Votre fichier Berksfile doit ressembler à ce qui suit.

   ```
   source 'https://supermarket.chef.io
        cookbook 'chef-client'
        cookbook 'apache2', '~> 5.0.1'
        cookbook 'opsworks-audit', path: 'cookbooks/opsworks-audit', '~> 1.0.0'
   ```

   Tous les livres de recettes sont versionnés dans le fichier `metadata.rb` du livre de recettes. Chaque fois que vous modifiez un livre de recettes, vous devez modifier la version du livre de recettes se trouvant dans son fichier `metadata.rb`.

3. Exécutez la commande suivante pour télécharger et installer les livres de recettes dans le dossier `cookbooks` sur votre ordinateur local ou votre ordinateur de travail.

   ```
   berks vendor cookbooks
   ```

4. Exécutez la commande suivante pour télécharger les livres de recettes fournis sur votre serveur AWS OpsWorks for Chef Automate .

```
knife upload .
```

5. Exécutez la commande suivante pour vérifier l'installation du livre de recettes `opsworks-audit` en affichant la liste des livres de recettes actuellement disponibles sur le serveur.

```
knife cookbook list
```

6. Ajoutez des nœuds à votre serveur pour les gérer, si vous ne l'avez pas déjà fait. Vous pouvez automatiser l'association des nœuds en suivant les étapes décrites dans Ajouter des nœuds automatiquement AWS OpsWorks for Chef Automate ou ajouter des nœuds l'un après l'autre en suivant les étapes indiquées dans Ajouter des nœuds individuellement. Modifiez la liste d'exécution de vos nœuds afin d'ajouter le rôle que vous avez spécifié à l'étape 1, `opsworks-example-role`. Dans cet exemple, nous modifions l'attribut `RUN_LIST` dans le script `userdata` que vous utilisez pour automatiser l'association des nœuds.

```
RUN_LIST="role[opsworks-example-role]"
```

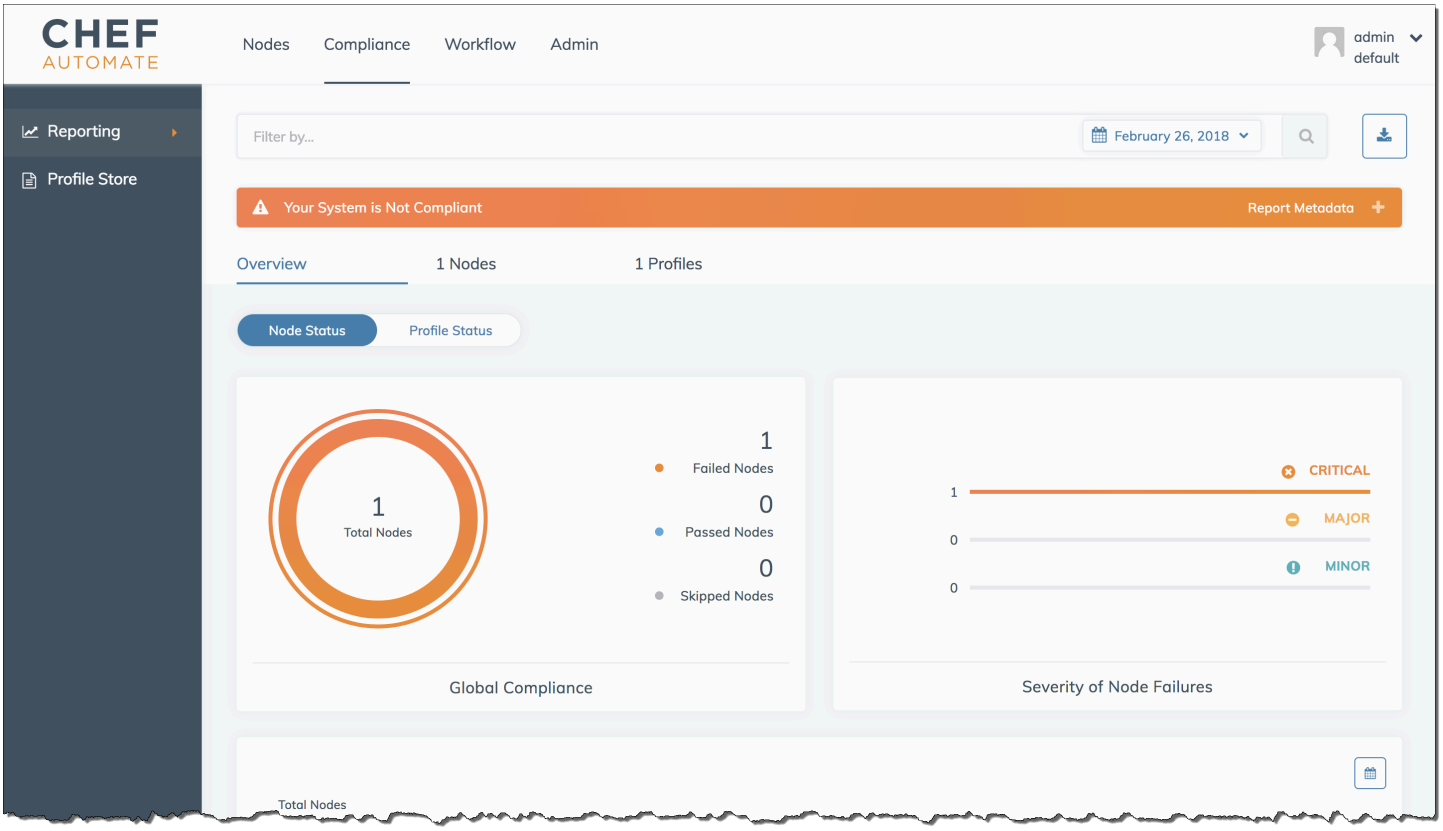
   Si vous avez ignoré l'étape 1 et que vous n'avez pas configuré le rôle, ajoutez les noms des recettes individuelles à la liste d'exécution. Enregistrez vos modifications et suivez les étapes ci-dessous Étape 3 : Création d'instances en utilisant un script d'association sans surveillance pour appliquer votre script de données utilisateur aux EC2 instances Amazon.

```
RUN_LIST="recipe[chef-client],recipe[apache2],recipe[opworks-audit]"
```

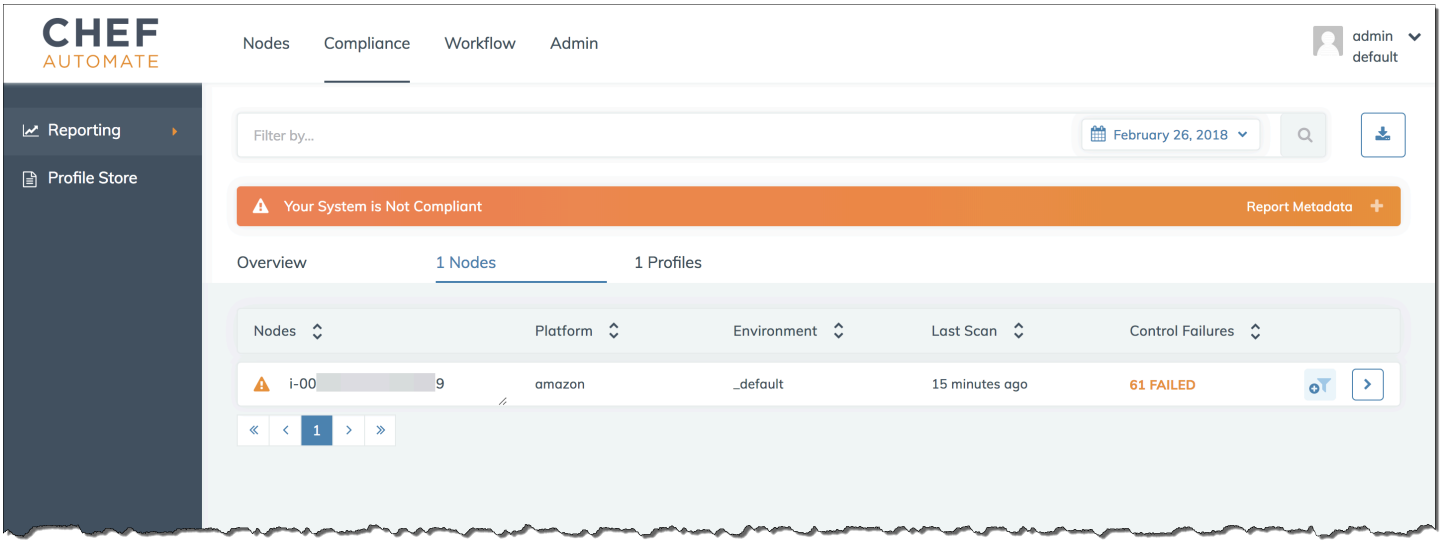
7. Une fois que vous avez mis à jour la liste d'exécution de vos nœuds, l'agent `chef-client` exécute les recettes spécifiées lors de son exécution suivante. Par défaut, cela se produit toutes les 1 800 secondes (30 minutes). Après l'exécution, vos résultats de conformité sont visibles dans le tableau de bord Chef Automate.

Exécution d'une analyse de conformité

Vous devriez voir les résultats de l'analyse de conformité dans le tableau de bord Chef Automate peu de temps après la première exécution du programme fantôme de l'agent, qui se produit une fois que vous avez configuré les listes d'exécution des nœuds.



Dans le tableau de bord Chef Automate, choisissez l'onglet Compliance (Conformité). Dans le panneau de navigation de gauche, choisissez Reporting (Génération de rapports). Choisissez l'onglet Profiles (Profils), sélectionnez Scan Results (Résultats de l'analyse), puis choisissez un nœud présentant des échecs d'analyse pour en savoir plus sur les règles ayant entraîné l'échec du nœud.



Généralement, les résultats d'analyse ne sont pas conformes, car les nouveaux nœuds ne satisfont pas encore à toutes les règles du profil de base DevSec SSH. Le DevSec Hardening Framework, un

projet communautaire, propose des livres de recettes pour résoudre les problèmes qui enfreignent les règles du profil de base DevSec SSH.

(Facultatif) Résolution des résultats non conformes

Le kit de démarrage comprend un livre de recettes open source que vous pouvez exécuter pour corriger les résultats non conformes résultant d'exécutions effectuées avec le profil de base DevSec SSH. `ssh-hardening`

> **Note**
>
> Le `ssh-hardening` livre de recettes apporte des modifications à vos nœuds pour les mettre en conformité avec les règles de base DevSec SSH. Avant d'exécuter ce livre de recettes sur un nœud de production, consultez les détails du profil de base DevSec SSH dans la console Chef Automate pour comprendre les violations des règles ciblées par le livre de recettes. Consultez les informations sur le livre de recettes open source `ssh-hardening` avant de l'exécuter sur un nœud de production.

Pour exécuter le livre de recettes **`ssh-hardening`**

1. Dans un éditeur de texte, ajoutez le `ssh-hardening`livre de recettes à votre fichier Berksfile. Votre fichier Berksfile doit ressembler à ce qui suit.

   ```
   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. Exécutez les commandes suivantes pour télécharger le livre de recettes `ssh-hardening` dans votre dossier de livres de recettes local, puis chargez-le sur votre serveur AWS OpsWorks for Chef Automate .

   ```
   berks vendor cookbooks
   knife upload .
   ```

3. Ajoutez la recette `ssh-hardening` à votre liste d'exécution de nœuds comme décrit dans les étapes 1 et 6 de Pour installer et configurer le livre de recettes `opsworks-audit`.

Si vous mettez à jour le fichier `opsworks-example-role.rb`, chargez vos modifications sur votre serveur en exécutant la commande suivante :

```
knife upload .
```

Si vous mettez directement à jour la liste d'exécution, chargez vos modifications en exécutant la commande suivante. Le nom de nœud est généralement l'ID de l'instance.

```
knife node run_list add <node name> 'recipe[ssh-hardening]'
```

4. Étant donné que vous utilisez le livre de recettes `chef-client`, votre nœud effectue des vérifications à intervalles réguliers (par défaut, toutes les 30 minutes). Lors du prochain enregistrement, le `ssh-hardening` livre de recettes s'exécute et contribue à améliorer la sécurité des nœuds afin de respecter les règles du profil DevSec SSH Baseline.
5. Une fois l'exécution initiale du livre de recettes `ssh-hardening` terminée, attendez 30 minutes afin qu'une analyse de conformité s'exécute à nouveau. Affichez les résultats dans le tableau de bord Chef Automate. Les résultats non conformes survenus lors de l'exécution initiale de l'analyse DevSec SSH Baseline doivent être résolus.

Mises à jour de la conformité

Sur un AWS OpsWorks for Chef Automate serveur, les fonctionnalités de conformité sont mises à jour automatiquement par la maintenance planifiée de votre système. Au fur et à mesure que des versions mises à jour de Chef Automate, Chef Infra Server et Chef InSpec seront disponibles pour votre AWS OpsWorks for Chef Automate serveur, vous devrez peut-être vérifier et mettre à jour les versions prises en charge du livre de recettes Audit et de la InSpec gem Chef qui s'exécutent sur votre serveur. Les profils que vous avez déjà installés sur votre AWS OpsWorks for Chef Automate serveur ne sont pas mis à jour dans le cadre de la maintenance.

Profils de conformité personnalisés et de la communauté

Actuellement, Chef comprend plus de 100 profils d'analyse de conformité. Vous pouvez ajouter des profils personnalisés et créés par la communauté à cette liste, puis télécharger et exécuter des analyses de conformité en fonction de ces profils, comme vous le feriez pour les profils déjà inclus. Des profils de conformité créés par la communauté sont disponibles à partir du site web

Chef Supermarket. Les profils personnalisés sont des programmes Ruby qui incluent un dossier de commandes spécifiant vos règles d'analyse.

consultez aussi

- Annonce sur le blog concernant Chef Compliance
- Formation en ligne à la conformité Chef Automate
- InSpecSite web du chef
- InSpec Tutoriels de chef

Dissocier un nœud d'un serveur AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section décrit comment dissocier ou supprimer un nœud géré de la gestion par un AWS OpsWorks for Chef Automate serveur. Cette opération est effectuée sur la ligne de commande ; vous ne pouvez pas dissocier les nœuds dans la console AWS OpsWorks for Chef Automate de gestion. Actuellement, l' AWS OpsWorks for Chef Automate API n'autorise pas la suppression par lots de plusieurs nœuds. La commande de cette section dissocie un nœud à la fois.

Nous vous recommandons de dissocier des nœuds d'un serveur Chef avant de supprimer le serveur pour que les nœuds continuent de fonctionner sans tenter de se reconnecter au serveur. Pour ce faire, exécutez la `disassociate-node` AWS CLI commande.

Pour dissocier des nœuds

1. Dans le AWS CLI, exécutez la commande suivante pour dissocier les nœuds. *`Node_name`*est le nom du nœud que vous souhaitez dissocier ; pour les EC2 instances Amazon, il s'agit de l'ID de l'instance. *`Server_name`*est le nom du serveur Chef dont vous souhaitez dissocier le nœud.

`--engine-attributes`indique votre `CHEF_AUTOMATE_ORGANIZATION` nom par défaut. Ces trois paramètres sont obligatoires.

Le paramètre `--region` n'est pas obligatoire, sauf si vous souhaitez dissocier un nœud d'un serveur de Chef qui n'est pas dans votre région par défaut.

```
aws opsworks-cm --region Region_name disassociate-node --node-
name Node_name --server-name Server_name --engine-attributes
 "Name=CHEF_AUTOMATE_ORGANIZATION,Value='default'"
```

Voici un exemple de commande.

```
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. Attendez qu'un message de réponse indique que la dissociation est terminée.

   Une fois que vous avez réussi à dissocier un nœud d'un AWS OpsWorks for Chef Automate serveur, il est possible qu'il soit toujours visible dans le tableau de bord de Chef Automate. Par défaut, Chef applique une période de conservation pour les informations d'état des nœuds et purge le nœud automatiquement au bout de quelques jours.

Pour plus d'informations sur la suppression d'un AWS OpsWorks for Chef Automate serveur, consultezSupprimer un AWS OpsWorks for Chef Automate serveur.

Rubriques connexes

Les articles de AWS blog suivants offrent plus d'informations sur l'association automatique de nœuds à votre serveur Chef Automate, à l'aide de groupes Auto Scaling ou au sein de plusieurs comptes.

- Utilisation d'AWS OpsWorks pour Chef Automate pour gérer les EC2 instances avec Auto Scaling
- OpsWorks pour Chef Automate — Démarrage automatique des nœuds dans différents comptes

Supprimer un AWS OpsWorks for Chef Automate serveur

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section explique comment supprimer un AWS OpsWorks for Chef Automate serveur. La suppression d'un serveur supprime également de ses événements, ses journaux et les livres de recettes qui ont été stockés sur le serveur. Les ressources de support (instance Amazon Elastic Compute Cloud, volume Amazon Elastic Block Store, etc.) sont également supprimées, ainsi que toutes les sauvegardes automatisées.

Même si la suppression d'un serveur ne supprime pas de nœuds, ces derniers ne sont plus gérés par le serveur supprimé et tenteront continuellement de se reconnecter. C'est pourquoi nous recommandons de dissocier les nœuds gérés avant de supprimer un serveur Chef. Dans cette version, vous pouvez dissocier les nœuds en exécutant une AWS CLI commande.

Étape 1 : Dissociation de nœuds gérés

Dissociez les nœuds du serveur Chef avant de supprimer le serveur pour que les nœuds continuent de fonctionner sans tenter de se reconnecter au serveur. Pour ce faire, exécutez la `disassociate-node` AWS CLI commande.

Pour dissocier des nœuds

1. Dans le AWS CLI, exécutez la commande suivante pour dissocier les nœuds. *`Server_name`*est le nom du serveur Chef dont vous souhaitez dissocier le nœud.

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

2. Attendez qu'un message de réponse indique que la dissociation est terminée.

Étape 2 : Suppression du serveur

1. Sur la vignette du serveur dans le tableau de bord, développez le menu Actions.
2. Choisissez Delete server (Supprimer le serveur).
3. Lorsque vous êtes invité à confirmer la suppression, choisissez Oui.

Réinitialiser les informations d'identification du tableau de bord Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez changer régulièrement le mot de passe avec lequel vous vous connectez au tableau de bord Chef Automate. Vous pouvez également utiliser les AWS CLI commandes Amazon EC2 Systems Manager présentées dans cette section pour modifier le mot de passe du tableau de bord Chef Automate si vous l'avez perdu. La commande que vous utilisez varie selon que votre serveur Chef Automate exécute la version 1 ou la version 2 de Chef Automate.

1. Pour renvoyer l'ID d'instance de votre serveur Chef, ouvrez la AWS Management Console page suivante.

   https://console.aws.amazon.com/ec2/v2/maison ? region= *region_of_your_server* #instances:search= - aws-opsworks-cm *server_name*

   Par exemple, pour un serveur Chef nommé MyChefServerdans la région USA Ouest (Oregon), l'URL de la console serait la suivante.

   https://console.aws.amazon.com/ec2/v2/maison ? region=us-west-2 #instances:search= - aws-opsworks-cm MyChefServer

   Prenez note de l'ID d'instance qui s'affiche dans la console ; vous en aurez besoin pour modifier votre mot de passe.

2. Pour réinitialiser le mot de passe de connexion au tableau de bord Chef Automate, exécutez l'une des AWS CLI commandes suivantes, selon que votre serveur exécute Chef Automate 1 ou Chef Automate 2. Remplacez-le *enterprise_name* par le nom de votre entreprise ou organisation, *user_name* par le nom d'utilisateur d'un administrateur du serveur, *new_password* par le mot de passe que vous souhaitez utiliser et *region_name* par la région dans laquelle se trouve votre serveur. Si vous ne spécifiez pas de nom d'entreprise, le nom de l'entreprise sera `default`. Par défaut, *enterprise_name* c'est `default` (il s'agit du nom de l'organisation qui est toujours provisionnée). Car*user_name*, crée AWS OpsWorks for Chef Automate uniquement un utilisateur nomméadmin. Prenez note du nouveau mot de passe et conservez-le dans un endroit sûr mais pratique.

   Pour 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
   ```

   Pour 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. Attendez le texte de sortie (dans ce cas, l'ID de la commande) pour montrer que le changement de mot de passe est terminé.

Journalisation des appels d' AWS OpsWorks for Chef Automate API avec AWS CloudTrail

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks for Chef Automate est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par une identité IAM, ou à un AWS service dans AWS OpsWorks for Chef Automate. CloudTrail capture tous les appels d'API AWS OpsWorks for Chef Automate sous forme d'événements, y compris les appels depuis la AWS OpsWorks for Chef Automate console et les appels de code vers le AWS OpsWorks for Chef Automate APIs. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris les événements pour AWS OpsWorks for Chef Automate. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite AWS OpsWorks for Chef Automate, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des détails supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

AWS OpsWorks for Chef Automate Informations dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité se produit dans AWS OpsWorks for Chef Automate, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre AWS compte. Pour plus d'informations, consultez la section Affichage des événements à l'aide de l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements pour AWS OpsWorks for Chef Automate, créez un parcours. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez :

- Présentation de la création d'un journal d'activité
- CloudTrail Services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

Toutes les AWS OpsWorks for Chef Automate actions sont enregistrées CloudTrail et documentées dans la référence de l'AWS OpsWorks for Chef Automate API. Par exemple, les appels aux CreateServerCreateBackup, et DescribeServersles actions génèrent des entrées dans les fichiers CloudTrail journaux.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec des informations d'identification d'utilisateur root ou IAM.
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour plus d'informations, consultez la section Élément userIdentity CloudTrail .

Comprendre les entrées du fichier AWS OpsWorks for Chef Automate journal

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal pour l' AWS OpsWorks for Chef Automate `CreateServer`action.

```
{"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"
}
```


Résolution des problèmes AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks for Chef Automate a atteint sa fin de vie le 5 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons aux clients existants de migrer vers Chef SaaS ou vers une solution alternative. Si vous avez des questions, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette rubrique présente certains AWS OpsWorks for Chef Automate problèmes courants et propose des solutions à ces problèmes.

Rubriques

- Conseils généraux de résolution des problèmes
- Résolution d'erreurs spécifiques
- Aide supplémentaire et support

Conseils généraux de résolution des problèmes

Si vous ne pouvez pas créer ou utiliser un serveur de Chef, vous pouvez afficher les messages d'erreur ou les journaux pour vous aider à résoudre le problème. Les tâches suivantes décrivent les emplacements généraux où démarrer lorsque vous voulez résoudre un problème lié à un serveur Chef. Pour plus d'informations sur des erreurs spécifiques et les solutions, consultez la section Résolution d'erreurs spécifiques de cette rubrique.

- Utilisez la AWS OpsWorks for Chef Automate console pour afficher les messages d'erreur si un serveur Chef ne démarre pas. Sur la page des détails du serveur Chef, les messages d'erreur liés au lancement et au fonctionnement du serveur sont affichés en haut de la page. Les erreurs peuvent provenir de AWS OpsWorks for Chef Automate AWS CloudFormation, ou d'Amazon EC2, des services utilisés pour créer un serveur Chef. Sur la page des détails, vous pouvez aussi afficher des événements se produisant sur un serveur en cours d'exécution, qui peut contenir des messages d'événement d'erreur.
- Pour résoudre EC2 les problèmes, connectez-vous à l'instance de votre serveur à l'aide de SSH et consultez les journaux. EC2 les journaux d'instance sont stockés dans le `/var/log/aws/opsworks-cm` répertoire. Ces journaux capturent les sorties de commande lors du AWS OpsWorks for Chef Automate lancement d'un serveur Chef.

Résolution d'erreurs spécifiques

Rubriques

- Le serveur est dans un état de perte de connexion
- Un nœud géré apparaît dans le tableau de bord Chef Automate dans la colonne Missing
- Impossible de créer un coffre Chef ; la commande knife vault échoue avec des erreurs
- La création du serveur échoue avec un message indiquant que la configuration demandée n'est actuellement pas prise en charge.
- Le serveur Chef ne reconnaît pas les noms d'organisation ajoutés dans le tableau de bord de Chef Automate.
- Impossible de créer l' EC2 instance Amazon du serveur
- Une erreur de rôle de service empêche la création du serveur
- Limite appliquée aux adresses IP Elastic dépassée
- Impossible de se connecter au tableau de bord Chef Automate

- Une association de nœud sans surveillance échoue
- Échec de la maintenance du système

Le serveur est dans un état de perte de connexion

Problème : l'état d'un serveur indique que la connexion est perdue.

Cause : Cela se produit le plus souvent lorsqu'une entité extérieure AWS OpsWorks apporte des modifications à un AWS OpsWorks for Chef Automate serveur ou à ses ressources de support. AWS OpsWorks ne peut pas se connecter aux serveurs Chef Automate en état de perte de connexion pour gérer les tâches de maintenance telles que la création de sauvegardes, l'application de correctifs du système d'exploitation ou la mise à jour de Chef Automate. Par conséquent, il est possible que votre serveur ne dispose pas de mises à jour importantes, soit exposé à des problèmes de sécurité ou ne fonctionne pas comme prévu.

Solution : essayez les étapes suivantes pour rétablir la connexion du serveur.

1. Assurez-vous que votre rôle de service dispose de toutes les autorisations requises.

    a. Sur la page Paramètres de votre serveur, dans Réseau et sécurité, choisissez le lien correspondant au rôle de service utilisé par le serveur. Cela ouvre le rôle de service à afficher dans la console IAM.

    b. Dans l'onglet Autorisations, vérifiez que cela `AWSOpsWorksCMServiceRole` figure dans la liste des politiques d'autorisations. Si elle n'est pas répertoriée, ajoutez la politique `AWSOpsWorksCMServiceRole` gérée manuellement au rôle.

    c. Dans l'onglet Relations de confiance, vérifiez que le rôle de service dispose d'une politique de confiance qui autorise le `opsworks-cm.amazonaws.com` service à assumer des rôles en votre nom. Pour plus d'informations sur l'utilisation des politiques de confiance avec les rôles, consultez Modifier un rôle (console) ou le billet du blog sur la AWS sécurité intitulé Comment utiliser les politiques de confiance avec les rôles IAM.

2. Assurez-vous que votre profil d'instance dispose de toutes les autorisations requises.

    a. Sur la page Paramètres de votre serveur, dans Réseau et sécurité, choisissez le lien correspondant au profil d'instance utilisé par le serveur. Cela ouvre le profil de l'instance pour qu'il soit affiché dans la console IAM.

    b. Dans l'onglet Autorisations, vérifiez cela `AmazonEC2RoleforSSM` et `AWSOpsWorksCMInstanceProfileRole` les deux figurent dans la liste des politiques

d'autorisations. Si l'une ou les deux ne figurent pas dans la liste, ajoutez ces politiques gérées manuellement au rôle.

c. Dans l'onglet Relations de confiance, vérifiez que le rôle de service dispose d'une politique de confiance qui autorise le `ec2.amazonaws.com` service à assumer des rôles en votre nom. Pour plus d'informations sur l'utilisation des politiques de confiance avec les rôles, consultez Modifier un rôle (console) ou le billet du blog sur la AWS sécurité intitulé Comment utiliser les politiques de confiance avec les rôles IAM.

3. Dans la EC2 console Amazon, assurez-vous que vous vous trouvez dans la même région que celle du AWS OpsWorks for Chef Automate serveur, puis redémarrez l' EC2 instance utilisée par votre serveur.

   a. Choisissez l' EC2 instance nommée `aws-opsworks-cm-instance-`*`server-name`*.

   b. Dans le menu État de l'instance, choisissez Redémarrer l'instance.

   c. Prévoyez jusqu'à 15 minutes pour que votre serveur redémarre et soit entièrement en ligne.

4. Dans la AWS OpsWorks for Chef Automate console, sur la page de détails du serveur, vérifiez que l'état du serveur est désormais sain.

Si l'état du serveur est toujours Perte de connexion après avoir effectué les étapes précédentes, essayez l'une des solutions suivantes.

- Remplacez le serveur en créant un nouveau et en supprimant l'original. Si les données du serveur actuel sont importantes pour vous, restaurez le serveur à partir d'une sauvegarde récente et vérifiez que les données sont à jour avant de supprimer le serveur d'origine qui ne répond pas.
- Contactez AWS le support.

Un nœud géré apparaît dans le tableau de bord Chef Automate dans la colonne Missing

Problème : Un nœud géré apparaît dans la colonne Missing (Manquant) du tableau de bord Chef Automate.

Cause : Lorsqu'un nœud ne s'est pas connecté au serveur Chef Automate depuis plus de 12 heures, et que `chef-client` ne peut pas s'exécuter sur le nœud, le nœud reprend l'état qu'il avait 12 heures auparavant et est transféré dans la colonne Missing (Manquant) du tableau de bord Chef Automate.

Solution : Vérifiez si le nœud est en ligne. Essayez de lancer `knife node show` *`node_name`* `--run-list` pour voir si `chef-client` peut s'exécuter sur le nœud, ou `knife node show -l` *`node_name`* pour afficher toutes les informations concernant le nœud. Le nœud est peut-être hors connexion ou déconnecté du réseau.

Impossible de créer un coffre Chef ; la commande **`knife vault`** échoue avec des erreurs

Problème : Vous essayez de créer un coffre sur votre serveur Chef Automate (par exemple, un coffre pour stocker des informations d'identification pour des nœuds Windows de jointure au domaine) en exécutant la commande `knife vault`. La commande renvoie un message d'erreur similaire à celui qui apparaît ci-dessous.

```
WARN: Auto inflation of JSON data is deprecated. Please pass in the class to inflate or
 use #edit_hash (CHEF-1)
at /opt/chefdk/embedded/lib/ruby/2.3.0/forwardable.rb:189:in `edit_data'.Please see
 https://docs.chef.io/deprecations_json_auto_inflate.html
for further details and information on how to correct this problem.
WARNING: pivotal not found in users, trying clients.
ERROR: ChefVault::Exceptions::AdminNotFound: FATAL: Could not find pivotal in users or
 clients!
```

L'utilisateur central n'est pas renvoyé lorsque vous exécutez la commande `knife user list` à distance, mais vous pouvez voir l'utilisateur central dans les résultats lorsque vous exécutez la commande `chef-server-ctl user-show` en local sur votre serveur Chef Automate. En d'autres termes, votre commande `knife vault` ne peut pas trouver l'utilisateur central mais vous savez qu'il existe.

Cause : Même si l'utilisateur central est considéré comme le super-utilisateur dans Chef et s'il dispose des autorisations complètes, il n'est membre d'aucune organisation, ni même de l'organisation `default` utilisée dans AWS OpsWorks for Chef Automate. La commande `knife user list` renvoie tous les utilisateurs figurant dans l'organisation actuelle de votre configuration Chef. La commande `chef-server-ctl user-show` renvoie tous les utilisateurs quelle que soit l'organisation, y compris l'utilisateur central.

Solution : Pour remédier à ce problème, ajoutez l'utilisateur central dans l'organisation par défaut en exécutant `knife opc`.

Tout d'abord, vous devez installer le plug-in knife-opc.

```
chef gem install knife-opc
```

Une fois que vous avez installé le plug-in, exécutez la commande suivante pour ajouter l'utilisateur central dans l'organisation par défaut.

```
knife opc org user add default pivotal
```

Vous pouvez vérifier que l'utilisateur central fait partie de l'organisation par défaut en exécutant à nouveau `knife user list`. `pivotal` devrait apparaître dans les résultats. Ensuite, réessayez d'exécuter `knife vault`.

La création du serveur échoue avec un message indiquant que la configuration demandée n'est actuellement pas prise en charge.

Problème : Vous essayez de créer un serveur Chef Automate, mais la création du serveur échoue avec un message d'erreur similaire à « The requested configuration is currently not supported. Please check the documentation for supported configurations. »

Cause : Un type d'instance non pris en charge a peut-être été spécifié pour le serveur Chef Automate. Si vous choisissez de créer le serveur Chef Automate dans un VPC disposant d'une location autre que celle par défaut, comme une location pour des instances dédiées, toutes les instances à l'intérieur du VPC spécifié doivent être également associées à une location dédiée ou d'hôte. Comme certains types d'instance, comme t2, ne sont disponibles qu'avec la location par défaut, le type d'instance du serveur Chef Automate peut ne pas être pris en charge par le VPC spécifié et la création du serveur échoue.

Solution : Si vous choisissez un VPC disposant d'une location autre que celle par défaut, utilisez un type d'instance m4 qui peut prendre en charge une location dédiée.

Le serveur Chef ne reconnaît pas les noms d'organisation ajoutés dans le tableau de bord de Chef Automate.

Problème : vous avez ajouté de nouveaux noms d'organisation de flux de travail dans le tableau de bord de Chef Automate ou spécifié une valeur `CHEF_AUTOMATE_ORGANIZATION` autre que `"default"` dans le script d'association sans surveillance des nœuds, mais l'association de nœuds échoue. Votre serveur AWS OpsWorks for Chef Automate ne reconnaît pas les nouveaux noms d'organisation.

Cause : les noms d'organisation du flux de travail et ceux du serveur Chef sont différents. Vous pouvez créer de nouvelles organisations du flux de travail dans le tableau de bord web de Chef Automate, mais pas de nouveaux noms d'organisation du serveur Chef. Vous pouvez utiliser le tableau de bord de Chef Automate uniquement pour afficher les organisations existantes du serveur Chef. Une nouvelle organisation créée dans le tableau de bord de Chef Automate est une organisation du flux de travail et elle n'est pas reconnue par le serveur Chef. Vous ne pouvez pas créer de nouveaux noms d'organisation en les spécifiant dans le script d'association des nœuds. Si vous faites référence à un nom d'organisation dans un script d'association de nœuds alors que l'organisation n'a pas été ajoutée au serveur Chef, l'association de nœuds échouera.

Solution : pour créer de nouvelles organisation reconnues sur le serveur Chef, utilisez la commande `knife opc org create` ou exécutez `chef-server-ctl org-create`.

Impossible de créer l' EC2 instance Amazon du serveur

Problème : La création du serveur a échoué avec un message d'erreur similaire au suivant : « Impossible de créer les ressources suivantes : [EC2Instance]. Failed to receive 1 resource signal(s) within the specified duration. »

Cause : Cela est probablement dû au fait que l' EC2instance n'a pas accès au réseau.

Solution : assurez-vous que l'instance dispose d'un accès Internet sortant et que l'agent de AWS service est en mesure d'émettre des commandes. Assurez-vous que le paramètre Résolution DNS est activé pour votre VPC (un VPC avec un seul sous-réseau public), et que le paramètre Activer l'adresse IP publique attribuée automatiquement est activé pour votre sous-réseau.

Une erreur de rôle de service empêche la création du serveur

Problème : la création du serveur échoue avec un message d'erreur indiquant « Non autorisé à exécuter sts : »AssumeRole.

Cause : Cela peut se produire lorsque le rôle de service vous utilisez n'a pas les autorisations appropriées pour créer un nouveau serveur.

Solution : ouvrez la AWS OpsWorks for Chef Automate console ; utilisez-la pour générer un nouveau rôle de service et un nouveau rôle de profil d'instance. Si vous préférez utiliser votre propre rôle de service, associez la politique AWSOpsWorks CMService Role au rôle. Vérifiez que opsworks-cm.amazonaws.com figure parmi les services concernés par les relations de confiance du rôle. Vérifiez que la politique gérée AWSOpsWorks Role est attachée au CMService rôle de service associé au serveur Chef.

Limite appliquée aux adresses IP Elastic dépassée

Problème : la création du serveur échoue avec un message d'erreur indiquant : « Impossible de créer les ressources suivantes : [EIP, EC2 instance]. Resource creation cancelled, the maximum number of addresses has been reached. »

Cause : Cette erreur a lieu lorsque votre compte a utilisé le nombre maximal d'adresses IP Elastic (EIP). La limite d'adresses IP Elastic est de cinq.

Solution : vous pouvez soit libérer les adresses EIP existantes, soit supprimer celles que votre compte n'utilise pas activement, soit contacter le Support AWS client pour augmenter le nombre d'adresses EIP associées à votre compte.

Impossible de se connecter au tableau de bord Chef Automate

Problème : Le tableau de bord Chef Automate affiche une erreur similaire à la suivante : « Demande d'origine croisée bloquée : la même politique d'origine interdit la lecture de la ressource distante lorsque la https://myserver-name.region.opsworks-cm.io/api/v0/e/default/verify-token. (Reason: CORS header 'Access-Control-Allow-Origin' missing)". The error can also be similar to "The User Id / Password com binaison saisie est incorrecte. »

Cause : Le tableau de bord Chef Automate définit explicitement le FQDN et n'accepte pas le relatif. URLs Pour le moment, vous ne pouvez pas vous connecter à l'aide de l'adresse IP Chef ; vous pouvez uniquement vous connecter en utilisant le nom DNS du serveur.

Solution : Connectez-vous au tableau de bord Chef Automate uniquement à l'aide de l'entrée de nom DNS du serveur Chef, pas avec son adresse IP. Vous pouvez également essayer de réinitialiser les informations d'identification du tableau de bord Chef Automate en exécutant une commande de l' AWS CLI , comme décrit dans Réinitialiser les informations d'identification du tableau de bord Chef Automate.

Une association de nœud sans surveillance échoue

Problème : l'association automatique ou non surveillée de nouveaux EC2 nœuds Amazon échoue. Les nœuds qui auraient dû être ajoutés au serveur Chef n'apparaissent pas dans le tableau de bord Chef Automate et ne sont pas répertoriés dans les résultats des commandes `knife client show` ou `knife node show`.

Cause : Cela peut se produire lorsqu'aucun rôle IAM n'est configuré en tant que profil d'instance permettant aux appels d'`opsworks-cm`API de communiquer avec de nouvelles EC2 instances.

Solution : associez à votre profil d' EC2 instance une politique permettant aux appels `AssociateNode` et à `DescribeNodeAssociationStatus` l'API de fonctionner EC2, comme décrit dansAjouter des nœuds automatiquement AWS OpsWorks for Chef Automate.

Échec de la maintenance du système

AWS OpsWorks CM effectue une maintenance hebdomadaire du système pour s'assurer que les dernières versions mineures de Chef Server et Chef Automate Server, y compris les mises à jour de sécurité, s'exécutent toujours sur un serveur AWS OpsWorks pour Chef Automate. Si, pour une raison quelconque, la maintenance du système échoue, vous en AWS OpsWorks CM informe. Pour plus d'informations sur la maintenance du système, consultezMaintenance du système dans AWS OpsWorks for Chef Automate.

Cette section décrit les causes possibles de l'échec et propose des solutions.

Rubriques

- Une erreur de rôle de service ou de profil d'instance empêche la maintenance du système

Une erreur de rôle de service ou de profil d'instance empêche la maintenance du système

Problème : la maintenance du système échoue avec un message d'erreur indiquant « Non autorisé à exécuter sts : AssumeRole » ou un message d'erreur similaire concernant les autorisations.

Cause : Cela peut se produire lorsque le rôle de service ou le profil d'instance que vous utilisez ne dispose pas des autorisations adéquates pour effectuer la maintenance du système sur le serveur.

Solution : Assurez-vous que votre rôle de service et votre profil d'instance disposent de toutes les autorisations requises.

1. Assurez-vous que votre rôle de service dispose de toutes les autorisations requises.

    a. Sur la page Paramètres de votre serveur, dans Réseau et sécurité, choisissez le lien correspondant au rôle de service utilisé par le serveur. Cela ouvre le rôle de service à afficher dans la console IAM.

    b. Dans l'onglet Autorisations, vérifiez qu'`AWSOpsWorksCMServiceRole`il est attaché au rôle de service. Si elle n'`AWSOpsWorksCMServiceRole`est pas répertoriée, ajoutez cette politique au rôle.

    c. Vérifiez que opsworks-cm.amazonaws.com figure parmi les services concernés par les relations de confiance du rôle. Pour plus d'informations sur l'utilisation des politiques de

confiance avec les rôles, consultez Modifier un rôle (console) ou le billet du blog sur la AWS sécurité intitulé Comment utiliser les politiques de confiance avec les rôles IAM.

2. Assurez-vous que votre profil d'instance dispose de toutes les autorisations requises.

   a. Sur la page Paramètres de votre serveur, dans Réseau et sécurité, choisissez le lien correspondant au profil d'instance utilisé par le serveur. Cela ouvre le profil de l'instance pour qu'il soit affiché dans la console IAM.

   b. Dans l'onglet Autorisations, vérifiez cela `AmazonEC2RoleforSSM` et `AWSOpsWorksCMInstanceProfileRole` les deux figurent dans la liste des politiques d'autorisations. Si l'une ou les deux ne figurent pas dans la liste, ajoutez ces politiques gérées manuellement au rôle.

   c. Dans l'onglet Relations de confiance, vérifiez que le rôle de service dispose d'une politique de confiance qui autorise le `ec2.amazonaws.com` service à assumer des rôles en votre nom. Pour plus d'informations sur l'utilisation des politiques de confiance avec les rôles, consultez Modifier un rôle (console) ou le billet du blog sur la AWS sécurité intitulé Comment utiliser les politiques de confiance avec les rôles IAM.

Aide supplémentaire et support

Si vous ne voyez pas votre problème spécifique décrit dans cette rubrique, ou si vous avez essayé les suggestions de cette rubrique et que les problèmes persistent, consultez les forums AWS OpsWorks.

Vous pouvez également visiter le Centre de Support AWS. Le Centre de AWS support est le centre de création et de gestion des dossiers de AWS support. Le Centre de AWS support inclut également des liens vers d'autres ressources utiles, telles que des forums, des informations techniques FAQs, l'état de santé des services et AWS Trusted Advisor.

Sécurité dans la gestion des AWS OpsWorks configurations (CM)

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit cette notion par les termes sécurité du cloud et sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de conformitéAWS. Pour en savoir plus sur les programmes de conformité qui s'appliquent à la gestion de configuration d' AWS OpsWorks , veuillez consulter Services AWS concernés par le programme de conformité.
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de AWS OpsWorks CM. Les rubriques suivantes expliquent comment configurer AWS OpsWorks CM pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres services AWS qui vous aident à surveiller et à sécuriser vos ressources AWS OpsWorks CM.

Rubriques

- Protection des données dans AWS OpsWorks CM
- Chiffrement des données
- Identity and Access Management pour AWS OpsWorks CM
- Confidentialité du trafic inter-réseaux
- Journalisation et surveillance dans AWS OpsWorks CM
- Validation de la conformité pour la gestion de configuration d' AWS OpsWorks
- Résilience dans le AWS OpsWorks CM

- Sécurité de l'infrastructure dans la gestion de configuration d' AWS OpsWorks
- Analyse de configuration et de vulnérabilité dans AWS OpsWorks CM
- Bonnes pratiques en matière de sécurité pour AWS OpsWorks CM

Protection des données dans AWS OpsWorks CM

Le modèle de responsabilité AWS partagée de s'applique à la protection des données dans AWS OpsWorks Configuration Management. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section Utilisation des CloudTrail sentiers dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez Norme FIPS (Federal Information Processing Standard) 140-3.

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec OpsWorks CM ou un autre utilisateur Services AWS à l'aide de la console AWS CLI, de l'API ou AWS SDKs. Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Les noms des serveurs OpsWorks CM ne sont pas chiffrés.

OpsWorks CM collecte les données clients suivantes dans le cadre de la création et de la maintenance de vos AWS OpsWorks for Puppet Enterprise serveurs AWS OpsWorks for Chef Automate et serveurs.

- OpsWorks Pour Puppet Enterprise, nous collectons les clés privées que Puppet Enterprise utilise pour permettre la communication entre votre Puppet Master et les nœuds gérés.
- En AWS OpsWorks for Chef Automate effet, nous collectons des clés privées pour les certificats que vous associez au service si vous utilisez un domaine personnalisé. La clé privée que vous fournissez lorsque vous créez un serveur Chef Automate avec un domaine personnalisé est transmise à votre serveur.

OpsWorks Les serveurs CM stockent votre code de configuration, tel que les livres de recettes Chef ou les modules Puppet Enterprise. Bien que ce code soit stocké dans les sauvegardes du serveur, il AWS n'y a pas accès. Ce contenu est crypté et seuls les administrateurs de votre AWS compte peuvent y accéder. Nous vous recommandons de sécuriser votre code de configuration Chef ou Puppet en utilisant les protocoles recommandés pour vos référentiels sources. Par exemple, vous pouvez limiter les autorisations aux référentiels situés sur le site Web ou suivre les directives du GitHub site Web pour sécuriser les GitHub référentiels. AWS CodeCommit

OpsWorks CM n'utilise pas le contenu fourni par le client pour maintenir le service ou conserver les journaux des clients. Les journaux relatifs à vos serveurs OpsWorks CM sont stockés dans votre compte, dans des compartiments Amazon S3. Les adresses IP des utilisateurs qui se connectent à vos serveurs OpsWorks CM sont enregistrées par AWS.

Intégration avec AWS Secrets Manager

À compter du 3 mai 2021, lorsque vous créez un nouveau serveur dans OpsWorks CM, OpsWorks CM stocke les secrets du serveur dans AWS Secrets Manager. Pour les nouveaux serveurs, les attributs suivants sont stockés sous forme de secrets dans Secrets Manager.

- Serveur Chef Automate
  - Clé privée HTTPS (uniquement les serveurs qui n'utilisent pas de domaine personnalisé)
  - Mot de passe administratif Chef Automate (CHEF_AUTOMATE_ADMIN_PASSWORD)
- Puppet Enterprise Master
  - Clé privée HTTPS (uniquement les serveurs qui n'utilisent pas de domaine personnalisé)
  - Mot de passe administratif de Puppet (PUPPET_ADMIN_PASSWORD)
  - Télécommande Puppet r10k (PUPPET_R10K_REMOTE)

Pour les serveurs existants qui n'utilisent pas de domaine personnalisé, le seul secret stocké dans Secrets Manager, pour les serveurs Chef Automate et Puppet Enterprise, est la clé privée HTTPS, car elle est générée lors de la maintenance automatique hebdomadaire du système.

OpsWorks CM stocke automatiquement les secrets dans Secrets Manager, et ce comportement n'est pas configurable par l'utilisateur.

Chiffrement des données

AWS OpsWorks CM chiffre les sauvegardes des serveurs et les communications entre les AWS utilisateurs autorisés et leurs serveurs AWS OpsWorks CM. Toutefois, les volumes Amazon EBS racine des serveurs AWS OpsWorks CM ne sont pas chiffrés.

Chiffrement au repos

AWS OpsWorks Les sauvegardes du serveur CM sont cryptées. Toutefois, les volumes Amazon EBS racine des serveurs AWS OpsWorks CM ne sont pas chiffrés. Ceci n'est pas configurable par l'utilisateur.

Chiffrement en transit

AWS OpsWorks CM utilise le protocole HTTP avec le cryptage TLS. AWS OpsWorks CM utilise par défaut des certificats auto-signés pour approvisionner et gérer les serveurs, si aucun certificat signé

n'est fourni par les utilisateurs. Nous vous recommandons d'utiliser un certificat signé par une autorité de certification.

Gestion des clés

AWS Key Management Service les clés gérées par le client et les clés gérées par AWS ne sont actuellement pas prises en charge par AWS OpsWorks CM.

Identity and Access Management pour AWS OpsWorks CM

AWS Identity and Access Management (IAM) est un AWS service qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources OpsWorks CM. IAM est un AWS service que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- Public ciblé
- Authentification avec des identités
- Gestion des accès à l'aide de politiques
- Comment fonctionne AWS OpsWorks CM avec IAM
- AWS OpsWorks Exemples de politiques basées sur l'identité CM
- Résolution des problèmes liés à AWS OpsWorks CM Identity and Access
- AWS politiques gérées pour la gestion AWS OpsWorks de la configuration
- Prévention interservices confuse des adjoints dans AWS OpsWorks CM

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez dans OpsWorks CM.

Utilisateur du service : si vous utilisez le service OpsWorks CM pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de plus en plus de fonctionnalités OpsWorks CM pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si

vous ne pouvez pas accéder à une fonction dans OpsWorks CM, consultezRésolution des problèmes liés à AWS OpsWorks CM Identity and Access.

Administrateur de service — Si vous êtes responsable des ressources OpsWorks CM dans votre entreprise, vous avez probablement un accès complet à OpsWorks CM. C'est à vous de déterminer à quelles fonctionnalités et ressources OpsWorks CM les utilisateurs de votre service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser IAM avec OpsWorks CM, consultezComment fonctionne AWS OpsWorks CM avec IAM.

Administrateur IAM : si vous êtes administrateur IAM, vous souhaiterez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à OpsWorks CM. Pour consulter des exemples de politiques basées sur l'identité OpsWorks CM que vous pouvez utiliser dans IAM, consultez. AWS OpsWorks Exemples de politiques basées sur l'identité CM

Authentification avec des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section Comment vous connecter à votre compte Compte AWS dans le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour

signer des demandes vous-même, consultez AWS Signature Version 4 pour les demandes d'API dans le Guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour plus d'informations, consultez Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Authentification multifactorielle AWS dans IAM dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur racine, consultez Tâches nécessitant les informations d'identification de l'utilisateur racine dans le Guide de l'utilisateur IAM.

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme telles que des mots de passe et des clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons d'effectuer une rotation des clés d'accès. Pour plus d'informations, consultez Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification dans le Guide de l'utilisateur IAM.

Un groupe IAM est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez nommer un groupe IAMAdminset lui donner les autorisations nécessaires pour administrer les ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Cas d'utilisation pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

> **Warning**
>
> Les utilisateurs IAM disposent d'informations d'identification à long terme, ce qui présente un risque de sécurité. Pour atténuer ce risque, nous vous recommandons de ne fournir à ces utilisateurs que les autorisations dont ils ont besoin pour effectuer la tâche et de supprimer ces autorisations lorsqu'elles ne sont plus nécessaires.

Rôles IAM

Un rôle IAM est une identité au sein de vous Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Pour assumer temporairement un rôle IAM dans le AWS Management Console, vous pouvez passer d'un rôle d'utilisateur à un rôle IAM (console). Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez Méthodes pour endosser un rôle dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré : pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .
- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.

- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
  - Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.
  - Rôle de service : il s'agit d'un rôle IAM attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.
  - Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.
- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une EC2 instance et qui envoient des demandes AWS CLI d' AWS API. Cela est préférable au stockage des clés d'accès dans l' EC2 instance. Pour attribuer un AWS rôle à une EC2 instance et le rendre disponible pour toutes ses applications, vous devez créer un profil d'instance attaché à l'instance.

Un profil d'instance contient le rôle et permet aux programmes exécutés sur l' EC2 instance d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez Utiliser un rôle IAM pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon dans le guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

OpsWorks CM prend en charge les politiques personnalisées que vous créez dans IAM et que vous associez à des utilisateurs, à des rôles ou à des groupes.

Politiques basées sur une ressource

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

OpsWorks CM ne prend pas en charge les politiques basées sur les ressources.

Listes de contrôle d'accès (ACLs)

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et AWS WAF Amazon VPC sont des exemples de services compatibles. ACLs Pour en savoir plus ACLs, consultez la présentation de la liste de contrôle d'accès (ACL) dans le guide du développeur Amazon Simple Storage Service.

OpsWorks CM n'en utilise pas ACLs.

Autres types de politique

OpsWorks CM ne prend pas en charge les autres types de politiques suivants.

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limites d'autorisations — Une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations obtenues représentent la combinaison des politiques basées sur l'identité de l'entité et de ses limites d'autorisations. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCPs) : SCPs politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée plusieurs AWS comptes détenus par votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer des politiques de contrôle des services (SCPs) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les Organizations SCPs, voir How SCPs Work dans le guide de AWS Organizations l'utilisateur.
- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez Politiques de session dans le Guide de l'utilisateur IAM.

Types de politique multiple

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Comment fonctionne AWS OpsWorks CM avec IAM

Avant d'utiliser IAM pour gérer l'accès à AWS OpsWorks CM, vous devez connaître les fonctionnalités IAM disponibles avec AWS OpsWorks CM. Pour obtenir une vue d'ensemble de la façon dont AWS OpsWorks CM et les autres AWS services fonctionnent avec IAM, consultez la section AWS Services compatibles avec IAM dans le Guide de l'utilisateur d'IAM.

Rubriques

- AWS OpsWorks Politiques basées sur l'identité CM
- AWS OpsWorks Politiques basées sur le CM et les ressources
- Autorisation basée sur les balises AWS OpsWorks CM
- AWS OpsWorks Rôles CM IAM

AWS OpsWorks Politiques basées sur l'identité CM

Avec les politiques basées sur l'identité IAM, vous pouvez spécifier les actions et les ressources autorisées ou refusées ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. AWS OpsWorks CM prend en charge des actions, des ressources et des clés de condition spécifiques. Pour en savoir plus sur tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Dans AWS OpsWorks CM, vous pouvez associer une déclaration de politique personnalisée à un utilisateur, à un rôle ou à un groupe.

Actions

L'élément `Action` d'une stratégie basée sur une identité IAM décrit les actions spécifiques qui seront autorisées ou refusées par la stratégie. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. L'action est utilisée dans une politique pour permettre d'effectuer l'opération associée.

Les actions de politique dans AWS OpsWorks CM utilisent le préfixe suivant avant l'action :`opsworks-cm:`. Par exemple, pour accorder à une personne l'autorisation de créer un serveur de gestion de configuration d' AWS OpsWorks à l'aide d'une opération d'API, vous incluez l'action `opsworks-cm:CreateServer` dans sa stratégie. Les déclarations de politique doivent inclure un `NotAction` élément `Action` ou. AWS OpsWorks CM définit son propre ensemble d'actions décrivant les tâches que vous pouvez effectuer avec ce service.

Pour spécifier plusieurs actions dans une seule déclaration, séparez-les par des virgules comme suit :

```
"Action": [
      "opsworks-cm:action1",
      "opsworks-cm:action2"
```

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Describe`, incluez l'action suivante :

```
"Action": "opsworks-cm:Describe*"
```

Lorsque vous utilisez des caractères génériques pour autoriser plusieurs actions dans une déclaration de stratégie, veillez à autoriser ces actions uniquement pour les services ou les utilisateurs autorisés.

Pour consulter la liste des actions AWS OpsWorks CM, consultez la section Actions, ressources et clés de condition pour AWS OpsWorks dans le guide de l'utilisateur IAM.

Ressources

L'élément `Resource` précise les objets auxquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Vous spécifiez une ressource à l'aide d'un ARN ou du caractère générique (*) pour indiquer que l'instruction s'applique à toutes les ressources.

Vous pouvez obtenir le numéro de ressource Amazon (ARN) d'un serveur AWS OpsWorks CM ou d'une sauvegarde en exécutant les opérations d'DescribeBackupsAPI DescribeServersou en basant les politiques au niveau des ressources sur ces ressources.

Une ressource de serveur AWS OpsWorks CM possède un ARN au format suivant :

```
arn:aws:opsworks-cm:{Region}:${Account}:server/${ServerName}/${UniqueId}
```

Une ressource de sauvegarde AWS OpsWorks CM possède un ARN au format suivant :

```
arn:aws:opsworks-cm:{Region}:${Account}:backup/${ServerName}-{Date-and-Time-Stamp-of-Backup}
```

Pour plus d'informations sur le format de ARNs, consultez Amazon Resource Names (ARNs) et AWS Service Namespaces.

Par exemple, pour spécifier le serveur Chef Automate `test-chef-automate` dans votre déclaration, utilisez l'ARN suivant :

```
"Resource": "arn:aws:opsworks-cm:us-west-2:123456789012:server/test-chef-automate/EXAMPLE-d1a2bEXAMPLE"
```

Pour spécifier tous les serveurs AWS OpsWorks CM appartenant à un compte spécifique, utilisez le caractère générique (*) :

```
"Resource": "arn:aws:opsworks-cm:us-west-2:123456789012:server/*"
```

L'exemple suivant indique une sauvegarde du serveur AWS OpsWorks CM en tant que ressource :

```
"Resource": "arn:aws:opsworks-cm:us-west-2:123456789012:backup/test-chef-automate-server-2018-05-20T19:06:12.399Z"
```

Certaines actions AWS OpsWorks CM, telles que celles relatives à la création de ressources, ne peuvent pas être effectuées sur une ressource spécifique. Dans ces cas-là, vous devez utiliser le caractère générique (*).

```
"Resource": "*"
```

De nombreuses actions d'API nécessitent plusieurs ressources. Pour spécifier plusieurs ressources dans une seule instruction, séparez-les ARNs par des virgules.

```
"Resource": [
      "resource1",
      "resource2"
```

Pour consulter la liste des types de ressources AWS OpsWorks CM et leurs caractéristiques ARNs, consultez la section Actions, ressources et clés de condition pour AWS OpsWorks CM dans le

guide de l'utilisateur IAM. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez la section Actions, ressources et clés de condition pour AWS OpsWorks CM dans le guide de l'utilisateur IAM.

Clés de condition

AWS OpsWorks CM ne dispose pas de clés contextuelles spécifiques au service pouvant être utilisées dans l'`Condition`élément des déclarations de politique. Pour obtenir la liste des clés de contexte globales disponibles pour tous les services, consultez la section Clés de contexte de condition AWS globale dans le manuel de référence des politiques IAM. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, comme égal ou inférieur, pour faire correspondre la condition de la stratégie aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez autoriser un utilisateur à accéder à une ressource uniquement si celle-ci porte le nom de l'utilisateur. Pour plus d'informations, consultez Éléments des politiques IAM : variables et balises dans le Guide de l'utilisateur IAM.

Exemples

Pour consulter des exemples de politiques basées sur l'identité AWS OpsWorks CM, consultez. AWS OpsWorks Exemples de politiques basées sur l'identité CM

AWS OpsWorks Politiques basées sur le CM et les ressources

AWS OpsWorks CM ne prend pas en charge les politiques basées sur les ressources.

Les stratégies basées sur les ressources sont des documents de stratégie JSON qui spécifient les actions qu'un mandataire spécifié peut effectuer sur une ressource et dans quelles conditions.

Autorisation basée sur les balises AWS OpsWorks CM

Vous pouvez associer des balises aux ressources AWS OpsWorks CM ou transmettre des balises dans une demande adressée à AWS OpsWorks CM. Pour contrôler l'accès basé sur des balises, vous devez fournir les informations des balises dans l'élément de condition d'une stratégie utilisant les clés de condition `aws:RequestTag/key-name` ou `aws:TagKeys`. Pour plus d'informations sur le balisage des ressources AWS OpsWorks CM, consultez Utilisation des balises sur les AWS OpsWorks for Chef Automate ressources ou Utilisation des balises sur les AWS OpsWorks for Puppet Enterprise ressources consultez ce guide.

AWS OpsWorks Rôles CM IAM

Un rôle IAM est une entité de votre AWS compte qui possède des autorisations spécifiques.

AWS OpsWorks CM utilise deux rôles :

- Rôle de service qui accorde au service AWS OpsWorks CM les autorisations nécessaires pour travailler dans le AWS compte d'un utilisateur. Si vous utilisez le rôle de service par défaut fourni par OpsWorks CM, le nom de ce rôle est`aws-opsworks-cm-service-role`.
- Rôle de profil d'instance qui permet au service AWS OpsWorks CM d'appeler l'API OpsWorks CM. Ce rôle donne accès à Amazon S3 et permet AWS CloudFormation de créer le serveur et le compartiment S3 pour les sauvegardes. Si vous utilisez le profil d'instance par défaut fourni par OpsWorks CM, le nom de ce rôle de profil d'instance est`aws-opsworks-cm-ec2-role`.

AWS OpsWorks CM n'utilise pas de rôles liés à un service.

Utilisation d'informations d'identification temporaires avec la gestion de configuration d' AWS OpsWorks

AWS OpsWorks CM prend en charge l'utilisation d'informations d'identification temporaires et hérite de cette fonctionnalité de AWS Security Token Service.

Vous pouvez utiliser des informations d'identification temporaires pour vous connecter à l'aide de la fédération, endosser un rôle IAM ou encore pour endosser un rôle intercompte. Vous obtenez des informations d'identification de sécurité temporaires en appelant des opérations d' AWS STS API telles que AssumeRoleou GetFederationToken.

Rôles liés à un service

AWS OpsWorks CM n'utilise pas de rôles liés à un service.

Les rôles liés aux AWS services permettent aux services d'accéder aux ressources d'autres services pour effectuer une action en votre nom. Les rôles liés à un service s'affichent dans votre compte IAM et sont la propriété du service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Rôles de service

Cette fonction permet à un service d'endosser une fonction du service en votre nom. Ce rôle autorise le service à accéder à des ressources d'autres services pour effectuer une action en votre nom. Les rôles de service s'affichent dans votre compte IAM et sont la propriété du compte. Cela signifie qu'un administrateur IAM peut modifier les autorisations associées à ce rôle. Toutefois, une telle action peut perturber le bon fonctionnement du service.

AWS OpsWorks CM utilise deux rôles :

- Rôle de service qui accorde au service AWS OpsWorks CM les autorisations nécessaires pour travailler dans le AWS compte d'un utilisateur. Si vous utilisez le rôle de service par défaut fourni par OpsWorks CM, le nom de ce rôle est`aws-opsworks-cm-service-role`.
- Rôle de profil d'instance qui permet au service AWS OpsWorks CM d'appeler l'API OpsWorks CM. Ce rôle donne accès à Amazon S3 et permet AWS CloudFormation de créer le serveur et le compartiment S3 pour les sauvegardes. Si vous utilisez le profil d'instance par défaut fourni par OpsWorks CM, le nom de ce rôle de profil d'instance est`aws-opsworks-cm-ec2-role`.

Choisir un rôle IAM dans CM AWS OpsWorks

Lorsque vous créez un serveur dans AWS OpsWorks CM, vous devez choisir un rôle pour permettre à AWS OpsWorks CM d'accéder à Amazon EC2 en votre nom. Si vous avez déjà créé un rôle de service, AWS OpsWorks CM vous propose une liste de rôles parmi lesquels choisir. OpsWorks CM peut créer le rôle pour vous, si vous ne le spécifiez pas. Il est important de choisir un rôle qui autorise l'accès au démarrage et à l'arrêt EC2 des instances Amazon. Pour plus d'informations, consultez Création d'un serveur Chef Automate ou Créer un serveur maître Puppet Enterprise.

AWS OpsWorks Exemples de politiques basées sur l'identité CM

Par défaut, les utilisateurs ou les rôles ne sont pas autorisés à créer ou à modifier des ressources AWS OpsWorks CM. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l' AWS API AWS Management Console AWS CLI, ou. Un administrateur IAM doit créer des politiques IAM qui accordent aux identités IAM l'autorisation d'effectuer des opérations d'API spécifiques sur les

ressources spécifiées dont ils ont besoin. Il doit ensuite attacher ces stratégies aux utilisateurs ou aux groupes ayant besoin de ces autorisations.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez Création de politiques dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Dans AWS OpsWorks CM, vous pouvez attribuer la `AWSOpsWorksCMServiceRole` politique à un utilisateur pour lui permettre de créer et de gérer des serveurs Chef Automate ou Puppet Enterprise à l'aide du AWS Management Console ou AWS CLI.

Rubriques

- Bonnes pratiques en matière de politiques
- Autoriser les utilisateurs à afficher leurs propres autorisations
- Affichage des serveurs AWS OpsWorks CM en fonction des balises

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources OpsWorks CM dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation d'IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.

- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que AWS CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politiques avec IAM Access Analyzer dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Sécurisation de l'accès aux API avec MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Autoriser les utilisateurs à afficher leurs propres autorisations

Cet exemple montre comment créer une stratégie qui permet aux utilisateurs d'afficher les stratégies en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or 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": "*"
        }
    ]
}
```

Affichage des serveurs AWS OpsWorks CM en fonction des balises

Vous pouvez utiliser les conditions de votre politique basée sur l'identité pour contrôler l'accès aux serveurs AWS OpsWorks CM et aux sauvegardes en fonction de balises. Cet exemple montre comment vous pouvez créer une politique qui autorise l'affichage d'un serveur AWS OpsWorks CM. Toutefois, l'autorisation n'est accordée que si la balise du serveur AWS OpsWorks CM `0wner` a la valeur du nom d'utilisateur de cet utilisateur. Cette politique accorde également les autorisations nécessaires pour réaliser cette action sur la console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListServersInConsole",
            "Effect": "Allow",
```

```
            "Action": "opsworks-cm:DescribeServers",
            "Resource": "*"
        },
        {
            "Sid": "ViewServerIfOwner",
            "Effect": "Allow",
            "Action": "opsworks-cm:DescribeServers",
            "Resource": "arn:aws:opsworks-cm:region:master-account-ID:server/server-
name",
            "Condition": {
                "StringEquals": {"opsworks-cm:ResourceTag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

Vous pouvez attacher cette stratégie aux utilisateurs de votre compte. Si un utilisateur nommé `richard-roe` tente de consulter un serveur AWS OpsWorks CM, le serveur doit être étiqueté `Owner=richard-roe` ou`owner=richard-roe`. Dans le cas contraire, l'utilisateur se voit refuser l'accès. La clé de condition d'étiquette `Owner` correspond à la fois à `Owner` et à `owner`, car les noms de clé de condition ne sont pas sensibles à la casse. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Résolution des problèmes liés à AWS OpsWorks CM Identity and Access

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec IAM. Pour obtenir des informations de dépannage spécifiques à AWS OpsWorks CM, reportez-vous aux Résolution des problèmes AWS OpsWorks for Chef Automate sections etRésolution des problèmes OpsWorks pour Puppet Enterprise.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans AWS OpsWorks CM
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes ressources AWS OpsWorks CM

Je ne suis pas autorisé à effectuer une action dans AWS OpsWorks CM

S'il vous AWS Management Console indique que vous n'êtes pas autorisé à effectuer une action, vous devez contacter votre administrateur pour obtenir de l'aide. Votre administrateur est la personne qui vous a fourni vos informations de connexion.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur `mateojackson` essaie d'utiliser la console pour afficher les détails d'un serveur AWS OpsWorks CM, mais ne dispose pas des `opsworks-cm:DescribeServers` autorisations nécessaires.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 opsworks-cm:DescribeServers on resource: test-chef-automate-server
```

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses stratégies pour lui permettre d'accéder à la ressource `test-chef-automate-server` à l'aide de l'action `opsworks-cm:DescribeServers`.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur selon lequel vous n'êtes pas autorisé à exécuter l'action `iam:PassRole`, vous devez contacter votre administrateur pour obtenir de l'aide. Votre administrateur est la personne qui vous a fourni vos informations de connexion. Demandez à cette personne de mettre à jour vos politiques afin de vous permettre de transférer un rôle à OpsWorks CM.

Certains AWS services vous permettent de transmettre un rôle existant à ce service, au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans OpsWorks CM. Toutefois, l'action nécessite que le service ait des autorisations accordées par une fonction du service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, Mary demande à son administrateur de mettre à jour ses politiques pour lui permettre d'exécuter l'action `iam:PassRole`.

Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes ressources AWS OpsWorks CM

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- AWS OpsWorks CM permet d'accorder aux utilisateurs de plusieurs comptes l'accès à la gestion d'un serveur AWS OpsWorks CM.
- Pour savoir comment fournir un accès à vos ressources sur les AWS comptes que vous possédez, consultez la section Fournir un accès à un utilisateur IAM sur un autre AWS compte que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des AWS comptes tiers, consultez la section Fournir un accès aux AWS comptes détenus par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des politiques basées sur les ressources pour l'accès entre comptes, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.

AWS politiques gérées pour la gestion AWS OpsWorks de la configuration

Pour ajouter des autorisations aux utilisateurs, aux groupes et aux rôles, il est plus facile d'utiliser des politiques AWS gérées que de les rédiger vous-même. Il faut du temps et de l'expertise pour créer des politiques gérées par le client IAM qui ne fournissent à votre équipe que les autorisations dont elle a besoin. Pour démarrer rapidement, vous pouvez utiliser nos politiques AWS gérées. Ces politiques couvrent les cas d'utilisation courants et sont disponibles dans votre AWS compte. Pour plus d'informations sur les politiques AWS gérées, voir les politiques AWS gérées dans le guide de l'utilisateur IAM.

AWS les services maintiennent et mettent à jour les politiques AWS gérées. Vous ne pouvez pas modifier les autorisations dans les politiques AWS gérées. Les services ajoutent occasionnellement

des autorisations à une politique gérée par AWS pour prendre en charge de nouvelles fonctionnalités. Ce type de mise à jour affecte toutes les identités (utilisateurs, groupes et rôles) auxquelles la politique est attachée. Les services sont très susceptibles de mettre à jour une politique gérée par AWS quand une nouvelle fonctionnalité est lancée ou quand de nouvelles opérations sont disponibles. Les services ne suppriment pas les autorisations d'une politique AWS gérée. Les mises à jour des politiques n'endommageront donc pas vos autorisations existantes.

En outre, AWS prend en charge les politiques gérées pour les fonctions professionnelles qui couvrent plusieurs services. Par exemple, la politique ReadOnlyAccess AWS gérée fournit un accès en lecture seule à tous les AWS services et ressources. Lorsqu'un service lance une nouvelle fonctionnalité, il AWS ajoute des autorisations en lecture seule pour les nouvelles opérations et ressources. Pour obtenir la liste des politiques de fonctions professionnelles et leurs descriptions, consultez la page politiques gérées par AWS pour les fonctions de tâche dans le Guide de l'utilisateur IAM.

Politique gérée par AWS : **`AWSOpsWorksCMServiceRole`**

Vous pouvez les associer `AWSOpsWorksCMServiceRole` à vos entités IAM. OpsWorks CM associe également cette politique à un rôle de service qui permet à OpsWorks CM d'effectuer des actions en votre nom.

Cette politique accorde des *`administrative`* autorisations qui permettent aux administrateurs OpsWorks CM de créer, de gérer et de supprimer des serveurs et des sauvegardes OpsWorks CM.

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- `opsworks-cm`— Permet aux principaux de supprimer des serveurs existants et de démarrer des opérations de maintenance.
- `acm`— Permet aux principaux de supprimer ou d'importer des certificats AWS Certificate Manager permettant aux utilisateurs de se connecter à un serveur OpsWorks CM.
- `cloudformation`— Permet à OpsWorks CM de créer et de gérer des AWS CloudFormation piles lorsque les principaux créent, mettent à jour ou suppriment des serveurs OpsWorks CM.
- `ec2`— Permet à OpsWorks CM de lancer, de provisionner, de mettre à jour et de résilier des instances Amazon Elastic Compute Cloud lorsque les principaux créent, mettent à jour ou suppriment des serveurs OpsWorks CM.
- `iam`— Permet à OpsWorks CM de créer les rôles de service nécessaires à la création et à la gestion des serveurs OpsWorks CM.

- `tag`— Permet aux principaux d'appliquer et de supprimer des balises sur les ressources OpsWorks CM, y compris les serveurs et les sauvegardes.
- `s3`— Permet à OpsWorks CM de créer des compartiments Amazon S3 pour stocker les sauvegardes de serveurs, de gérer les objets dans des compartiments S3 sur demande principale (par exemple, supprimer une sauvegarde) et de supprimer des compartiments.
- `secretsmanager`— Permet à OpsWorks CM de créer et de gérer les secrets de Secrets Manager, et d'appliquer ou de supprimer des balises dans les secrets.
- `ssm`— Permet à OpsWorks CM d'utiliser la commande Run Command de Systems Manager sur les instances qui sont des serveurs OpsWorks CM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket*"
            ],
            "Action": [
                "s3:CreateBucket",
                "s3:DeleteObject",
                "s3:DeleteBucket",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:PutBucketPolicy",
                "s3:PutObject",
                "s3:GetBucketTagging",
                "s3:PutBucketTagging"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Action": [
                "tag:UntagResources",
                "tag:TagResources"
            ]
        },
```

```
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Action": [
                "ssm:DescribeInstanceInformation",
                "ssm:GetCommandInvocation",
                "ssm:ListCommandInvocations",
                "ssm:ListCommands"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/aws:cloudformation:stack-name": "aws-opsworks-cm-
*"
                }
            },
            "Action": [
                "ssm:SendCommand"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:*::document/*",
                "arn:aws:s3:::amzn-s3-demo-bucket*"
            ],
            "Action": [
                "ssm:SendCommand"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Action": [
                "ec2:AllocateAddress",
```

```
                "ec2:AssociateAddress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateImage",
                "ec2:CreateSecurityGroup",
                "ec2:CreateSnapshot",
                "ec2:CreateTags",
                "ec2:DeleteSecurityGroup",
                "ec2:DeleteSnapshot",
                "ec2:DeregisterImage",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAddresses",
                "ec2:DescribeImages",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstances",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSnapshots",
                "ec2:DescribeSubnets",
                "ec2:DisassociateAddress",
                "ec2:ReleaseAddress",
                "ec2:RunInstances",
                "ec2:StopInstances"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "ec2:ResourceTag/aws:cloudformation:stack-name": "aws-opsworks-cm-
*"
                }
            },
            "Action": [
                "ec2:TerminateInstances",
                "ec2:RebootInstances"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:opsworks-cm:*:*:server/*"
            ],
```

```
            "Action": [
                "opsworks-cm:DeleteServer",
                "opsworks-cm:StartMaintenance"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:cloudformation:*:*:stack/aws-opsworks-cm-*"
            ],
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStackResources",
                "cloudformation:DescribeStacks",
                "cloudformation:UpdateStack"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iam::*:role/aws-opsworks-cm-*",
                "arn:aws:iam::*:role/service-role/aws-opsworks-cm-*"
            ],
            "Action": [
                "iam:PassRole"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "acm:DeleteCertificate",
                "acm:ImportCertificate"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "arn:aws:secretsmanager:*:*:opsworks-cm!aws-opsworks-cm-
secrets-*",
            "Action": [
                "secretsmanager:CreateSecret",
                "secretsmanager:GetSecretValue",
```

```
                "secretsmanager:UpdateSecret",
                "secretsmanager:DeleteSecret",
                "secretsmanager:TagResource",
                "secretsmanager:UntagResource"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DeleteTags",
            "Resource": [
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:elastic-ip/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        }
    ]
}
```

Politique gérée par AWS : **AWSOpsWorksCMInstanceProfileRole**

Vous pouvez les associer `AWSOpsWorksCMInstanceProfileRole` à vos entités IAM. OpsWorks CM associe également cette politique à un rôle de service qui permet à OpsWorks CM d'effectuer des actions en votre nom.

Cette politique accorde *`administrative`* des autorisations qui permettent aux EC2 instances Amazon utilisées comme serveurs OpsWorks CM d'obtenir des informations depuis AWS CloudFormation et AWS Secrets Manager de stocker des sauvegardes de serveurs dans des compartiments Amazon S3.

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- `acm`— Permet aux EC2 instances du serveur OpsWorks CM d'obtenir des certificats AWS Certificate Manager permettant aux utilisateurs de se connecter à un serveur OpsWorks CM.
- `cloudformation`— Permet aux EC2 instances du serveur OpsWorks CM d'obtenir des informations sur les AWS CloudFormation piles lors du processus de création ou de mise à jour des instances, et d'envoyer des signaux AWS CloudFormation concernant leur état.
- `s3`— Permet aux EC2 instances du serveur OpsWorks CM de télécharger et de stocker les sauvegardes du serveur dans des compartiments S3, d'arrêter ou d'annuler les téléchargements si nécessaire et de supprimer les sauvegardes des compartiments S3.

- `secretsmanager`— Permet aux EC2 instances du serveur OpsWorks CM d'obtenir les valeurs des secrets de Secrets Manager liés à OpsWorks CM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "cloudformation:DescribeStackResource",
                "cloudformation:SignalResource"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:ListMultipartUploadParts",
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket*",
            "Effect": "Allow"
        },
        {
            "Action": "acm:GetCertificate",
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "arn:aws:secretsmanager:*:*:opsworks-cm!aws-opsworks-cm-
secrets-*",
            "Effect": "Allow"
        }
    ]
}
```

OpsWorks Mises à jour des politiques AWS gérées par CM

Consultez les détails des mises à jour apportées aux politiques AWS gérées pour OpsWorks CM depuis que ce service a commencé à suivre ces modifications. Pour recevoir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS sur la page d'historique du document OpsWorks CM.

Modification	Description	Date
AWSOpsWorks CMInstance ProfileRole - Politique gérée mise à jour	OpsWorks CM a mis à jour la politique gérée qui permet aux EC2 instances utilisées comme serveurs OpsWorks CM de partager des informations avec CloudFormation Secrets Manager et de gérer les sauvegardes. La modification ajoute `opsworks-cm!` au nom de la ressource pour les secrets de Secrets Manager, de sorte que OpsWorks CM est autorisé à détenir les secrets.	23 avril 2021
AWSOpsWorks CMService Role - Politique gérée mise à jour	OpsWorks CM a mis à jour la politique gérée qui permet aux administrateurs OpsWorks CM de créer, de gérer et de supprimer des serveurs OpsWorks CM et des sauvegardes. La modification ajoute `opsworks-cm!` au nom de la ressource pour les secrets de Secrets Manager, de sorte que OpsWorks CM est autorisé à détenir les secrets.	23 avril 2021

Modification	Description	Date
OpsWorks CM a commencé à suivre les modifications	OpsWorks CM a commencé à suivre les modifications apportées AWS à ses politiques gérées.	23 avril 2021

Prévention interservices confuse des adjoints dans AWS OpsWorks CM

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner la confusion des adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé d'accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés de contexte de condition `aws:SourceAccount`globale `aws:SourceArn`et les clés contextuelles dans les politiques de ressources afin de limiter les autorisations qui AWS OpsWorks CM accordent un autre service à la ressource. Si la valeur `aws:SourceArn` ne contient pas l'ID du compte, tel qu'un ARN de compartiment Amazon S3, vous devez utiliser les deux clés de contexte de condition globale pour limiter les autorisations. Si vous utilisez les deux clés de contexte de condition globale et que la valeur `aws:SourceArn` contient l'ID de compte, la valeur `aws:SourceAccount` et le compte dans la valeur `aws:SourceArn` doivent utiliser le même ID de compte lorsqu'ils sont utilisés dans la même instruction de politique. Utilisez `aws:SourceArn` si vous souhaitez qu'une seule ressource soit associée à l'accès entre services. Utilisez `aws:SourceAccount` si vous souhaitez autoriser l'association d'une ressource de ce compte à l'utilisation interservices.

La valeur de `aws:SourceArn` doit être l'ARN d'un serveur OpsWorks CM Chef ou Puppet.

Le moyen le plus efficace de se protéger contre le problème de confusion des adjoints consiste à utiliser la clé de contexte de condition `aws:SourceArn` globale avec l'ARN complet du AWS OpsWorks CM serveur. Si vous ne connaissez pas l'ARN complet ou si vous spécifiez plusieurs serveurs ARNs, utilisez la clé de condition de contexte `aws:SourceArn` globale

avec des caractères génériques (*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:servicename:*:123456789012:*`.

La section suivante montre comment utiliser les touches de contexte de condition `aws:SourceAccount` globale `aws:SourceArn` et globale AWS OpsWorks CM pour éviter le problème de confusion des adjoints.

Empêchez les exploits d'adjoints confus dans AWS OpsWorks CM

Cette section décrit comment vous pouvez empêcher les exploits secondaires confus et inclut des exemples de politiques d'autorisation que vous pouvez associer au rôle IAM auquel vous accédez AWS OpsWorks CM. AWS OpsWorks CM Pour des raisons de sécurité, nous vous recommandons d'ajouter les clés de `aws:SourceAccount` condition `aws:SourceArn` et aux relations de confiance que votre rôle IAM entretient avec d'autres services. Les relations de confiance permettent AWS OpsWorks CM d'assumer un rôle pour effectuer des actions dans d'autres services nécessaires à la création ou à la gestion de vos AWS OpsWorks CM serveurs.

Pour modifier les relations de confiance afin d'ajouter **`aws:SourceArn`** et de **`aws:SourceAccount`** conditionner des clés

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation de gauche, choisissez Rôles.
3. Dans la zone de recherche, recherchez le rôle auquel vous souhaitez accéder AWS OpsWorks CM. Le rôle AWS géré est`aws-opsworks-cm-service-role`.
4. Sur la page Résumé du rôle, choisissez l'onglet Relations de confiance.
5. Dans l'onglet Trust relationships (Relations d'approbation), choisissez Edit trust relationship (Modifier la relation d'approbation).
6. Dans le document de stratégie, ajoutez au moins l'une des clés de `aws:SourceAccount` condition `aws:SourceArn` ou à la politique. `aws:SourceArn`À utiliser pour restreindre la relation de confiance entre les services interconnectés (tels que AWS Certificate Manager Amazon EC2) et avec AWS OpsWorks CM des AWS OpsWorks CM serveurs spécifiques, ce qui est plus restrictif. Ajoutez `aws:SourceAccount` pour restreindre la relation de confiance entre les services interservices et AWS OpsWorks CM les serveurs d'un compte spécifique, ce qui est moins restrictif. Voici un exemple. Notez que si vous utilisez les deux clés de condition, le compte IDs doit être le même.

```
{
```

```
  "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. Lorsque vous avez terminé d'ajouter des clés de condition, choisissez Mettre à jour la politique de confiance.

Vous trouverez ci-dessous d'autres exemples de rôles qui limitent l'accès aux AWS OpsWorks CM serveurs en utilisant `aws:SourceArn` et`aws:SourceAccount`.

Rubriques

- Exemple : accès aux AWS OpsWorks CM serveurs d'une région spécifique
- Exemple : ajout de plusieurs ARN de serveur à aws:SourceArn

Exemple : accès aux AWS OpsWorks CM serveurs d'une région spécifique

La déclaration de relation de confiance suivante permet d'accéder à tous AWS OpsWorks CM les serveurs de la région USA Est (Ohio) (`us-east-2`). Notez que la région est spécifiée dans la valeur ARN de`aws:SourceArn`, mais que la valeur de l'ID du serveur est un caractère générique (*).

```
{
  "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/*"
        }
      }
    }
  ]
}
```

Exemple : ajout de plusieurs ARN de serveur à **`aws:SourceArn`**

L'exemple suivant limite l'accès à un ensemble de deux AWS OpsWorks CM serveurs sous l'ID de compte 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"
           ]
       }
```

```
      }
    }
  ]
}
```

Confidentialité du trafic inter-réseaux

AWS OpsWorks CM utilise les mêmes protocoles de sécurité de transmission généralement utilisés par AWS : HTTPS ou HTTP avec cryptage TLS.

Journalisation et surveillance dans AWS OpsWorks CM

AWS OpsWorks CM enregistre toutes les actions de l'API dans CloudTrail. Pour plus d'informations, consultez les rubriques suivantes :

- Journalisation OpsWorks des appels d'API Puppet Enterprise avec AWS CloudTrail
- Journalisation des appels d' AWS OpsWorks for Chef Automate API avec AWS CloudTrail

Validation de la conformité pour la gestion de configuration d' AWS OpsWorks

AWS OpsWorks CM soutient les programmes et réglementations de conformité suivants :

- Payment Card Industry (PCI)
- Loi américaine Health Insurance Portability and Accountability Act (HIPAA) de 1996
- AWS Contrôles du système et de l'organisation (SOC) 1, 2 et 3
- Règlement général sur la protection des données (RGPD)

Des auditeurs tiers évaluent la sécurité et la conformité du AWS OpsWorks CM dans le cadre de multiples programmes de AWS conformité. Il s'agit notamment des certifications SOC, PCI, FedRAMP, HIPAA et d'autres.

Pour obtenir la liste des AWS services concernés par des programmes de conformité spécifiques, consultez la section Services AWS concernés par programme de conformité. Pour obtenir des informations générales, consultez Programmes de conformitéAWS.

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, consultez Téléchargement des rapports dans AWS Artifact.

Lorsque vous utilisez AWS OpsWorks CM, votre responsabilité en matière de conformité dépend de la sensibilité de vos données, des objectifs de conformité de votre entreprise et des lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides Quick Start de la sécurité et de la conformité : ces guides de déploiement traitent de considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de référence centrés sur la sécurité et la conformité dans AWS.
- Livre blanc sur l'architecture pour la sécurité et la conformité HIPAA — Ce livre blanc décrit comment les entreprises peuvent créer des applications conformes à la loi HIPAA. AWS
- AWS Ressources relatives à la conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- AWS Config— Ce AWS service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub— Ce AWS service fournit une vue complète de l'état de votre sécurité interne, AWS ce qui vous permet de vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.

Résilience dans le AWS OpsWorks CM

AWS OpsWorks CM permet des sauvegardes quotidiennes des serveurs par défaut lorsque vous créez un serveur. Les sauvegardes sont chiffrées et stockées dans un compartiment Amazon S3. Par défaut, ce compartiment n'est accessible qu'au compte qui a créé le serveur. Vous pouvez ajouter un accès au bucket pour d'autres utilisateurs ou configurer des sauvegardes entre régions dans Amazon S3 à votre guise. Chef et Puppet prennent en charge le chiffrement inter-régions, car les deux produits chiffrent le trafic entre votre serveur de gestion de configuration d' AWS OpsWorks et les nœuds gérés.

AWS OpsWorks CM ne prend pas en charge les configurations de haute disponibilité (HA).

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones

de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone de disponibilité à l'autre sans interruption. Les zones de disponibilité sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur la sauvegarde et la restauration de serveurs dans AWS OpsWorks CM, consultez les rubriques suivantes :

- Sauvegarder et restaurer un serveur OpsWorks pour Puppet Enterprise
- Sauvegarder et restaurer un AWS OpsWorks for Chef Automate serveur

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section Infrastructure AWS mondiale.

Sécurité de l'infrastructure dans la gestion de configuration d' AWS OpsWorks

En tant que service géré, AWS OpsWorks Configuration Management est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section Sécurité du AWS cloud. Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section Protection de l'infrastructure dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à OpsWorks CM via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

AWS OpsWorks CM ne prend pas en charge les liens privés ni les points de terminaison privés VPC.

AWS OpsWorks CM ne prend pas en charge les politiques basées sur les ressources. Pour plus d'informations, consultez la section AWS Services compatibles avec IAM dans le Guide de l'AWS Identity and Access Management utilisateur.

Analyse de configuration et de vulnérabilité dans AWS OpsWorks CM

AWS OpsWorks CM effectue des mises à jour périodiques du noyau et des mises à jour de sécurité du système d'exploitation qui s'exécute sur votre serveur AWS OpsWorks CM. Les utilisateurs peuvent définir une période pendant laquelle les mises à jour automatiques seront effectuées pendant une période maximale de deux semaines à compter de la date actuelle. AWS OpsWorks CM propose des mises à jour automatiques des versions mineures de Chef et Puppet Enterprise. Pour plus d'informations sur la configuration des mises à jour pour AWS OpsWorks for Chef Automate, consultez la section Maintenance du système (Chef) dans ce guide. Pour plus d'informations sur la configuration des mises à jour OpsWorks pour Puppet Enterprise, consultez la section Maintenance du système (Puppet) dans ce guide.

Bonnes pratiques en matière de sécurité pour AWS OpsWorks CM

AWS OpsWorks CM, comme tous les AWS services, propose des fonctionnalités de sécurité à prendre en compte lors de l'élaboration et de la mise en œuvre de vos propres politiques de sécurité. Les bonnes pratiques suivantes doivent être considérées comme des instructions générales et ne représentent pas une solution de sécurité complète. Étant donné que ces bonnes pratiques peuvent ne pas être appropriées ou suffisantes pour votre environnement, considérez-les comme des remarques utiles plutôt que comme des recommandations.

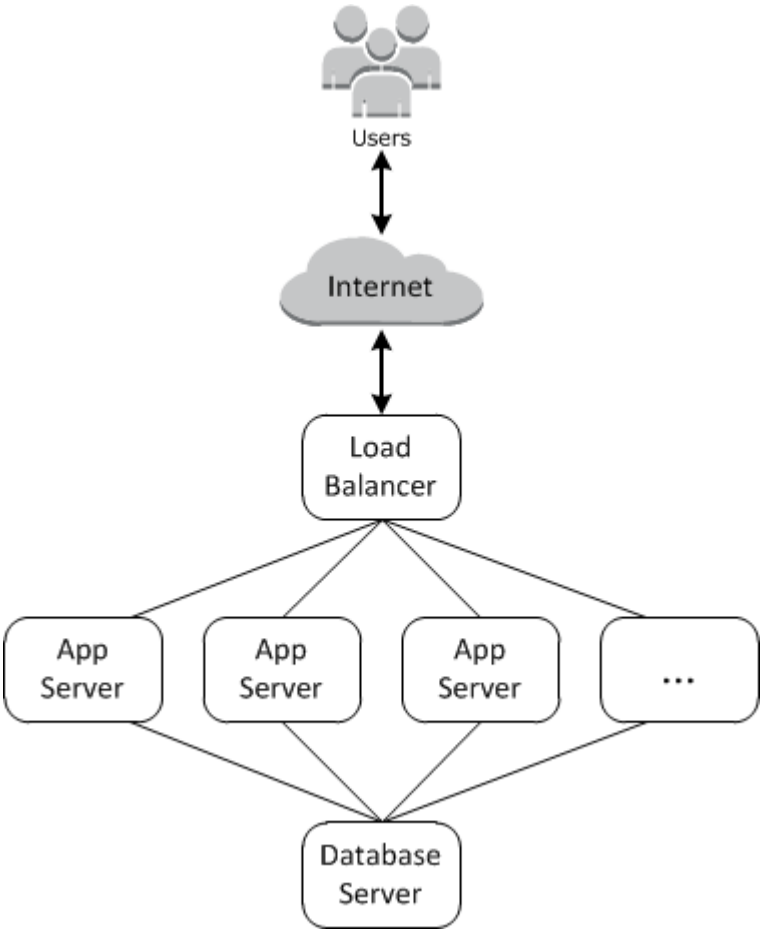
- Sécurisez votre kit de démarrage et vos identifiants de connexion téléchargés. Lorsque vous créez un nouveau serveur AWS OpsWorks CM ou que vous téléchargez un nouveau kit de démarrage et des informations d'identification depuis la console AWS OpsWorks CM, stockez ces éléments dans un emplacement sécurisé qui nécessite au moins un facteur d'authentification. Les informations d'identification fournissent un accès de niveau administrateur à votre serveur.
- Sécurisez votre code de configuration. Sécurisez votre code de configuration Chef ou Puppet (livres de recettes et modules) à l'aide des protocoles recommandés pour vos référentiels sources. Par exemple, vous pouvez limiter les autorisations aux référentiels situés sur le site Web ou suivre les directives du GitHub site Web pour sécuriser les GitHub référentiels. AWS CodeCommit

- Utilisez des certificats signés par une autorité de certification pour vous connecter aux nœuds. Bien que vous puissiez utiliser des certificats auto-signés lorsque vous enregistrez ou démarrez des nœuds sur votre serveur AWS OpsWorks CM, il est recommandé d'utiliser des certificats signés par une autorité de certification. Nous vous recommandons d'utiliser un certificat signé par une autorité de certification.
- Ne partagez pas les informations d'identification permettant de se connecter à la console de gestion Chef ou Puppet avec d'autres utilisateurs. Un administrateur doit créer des utilisateurs distincts pour chaque utilisateur des sites Web de la console Chef ou Puppet.
  - Gérer les utilisateurs dans Chef Automate
  - Gérer les utilisateurs dans Puppet Enterprise
- Configurez les sauvegardes et les mises à jour automatiques de maintenance du système. La configuration des mises à jour automatiques de maintenance sur votre serveur de gestion de configuration d' AWS OpsWorks permet de s'assurer que votre serveur exécute les mises à jour de sécurité les plus récentes du système d'exploitation. La configuration des sauvegardes automatiques facilite la reprise après sinistre et accélère la restauration en cas d'incident ou de panne. Limitez l'accès au compartiment Amazon S3 qui stocke les sauvegardes de votre serveur AWS OpsWorks CM ; n'accordez pas l'accès à tout le monde. Accordez un accès en lecture ou en écriture à d'autres utilisateurs individuellement selon les besoins, ou créez un groupe de sécurité dans IAM pour ces utilisateurs et attribuez l'accès au groupe de sécurité.
  - Maintenance du système (Chef)
  - Maintenance du système (Puppet)
  - Sauvegarder et restaurer un AWS OpsWorks for Chef Automate serveur
  - Sauvegarder et restaurer un serveur OpsWorks pour Puppet Enterprise
  - Création de votre premier groupe et utilisateur délégué IAM dans le Guide de l'utilisateur AWS Identity and Access Management
  - Bonnes pratiques de sécurité pour Amazon S3 dans le guide du développeur Amazon Simple Storage Service

AWS OpsWorks Piles

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

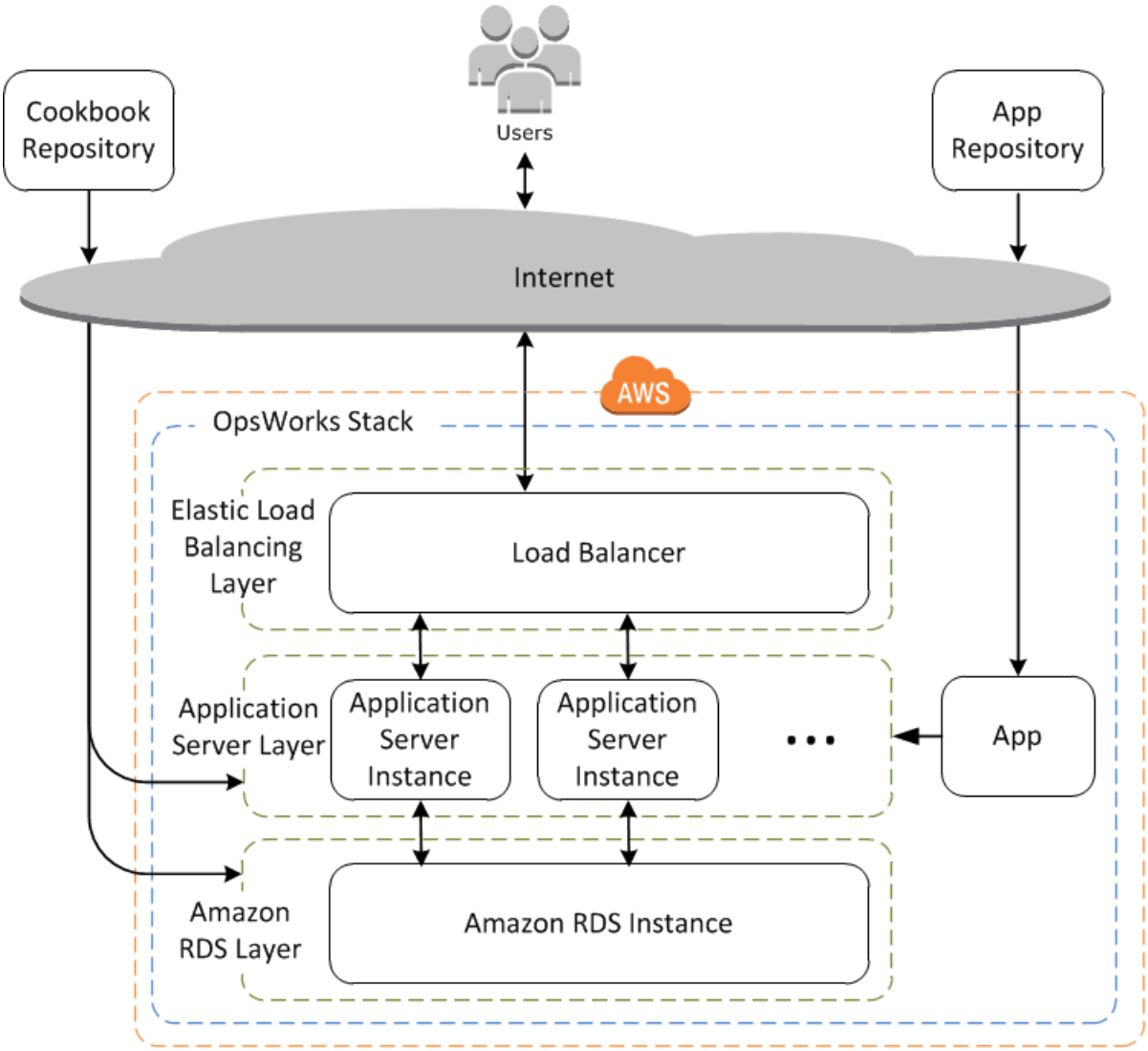
L'informatique basée sur le cloud implique généralement des groupes de ressources AWS, tels que EC2 des instances Amazon et des instances Amazon Relational Database Service (RDS), qui doivent être créées et gérées collectivement. Par exemple, une application web nécessite généralement des serveurs d'applications, des serveurs de base des données, des équilibreurs de charge, etc. Ce groupe d'instances est généralement appelé pile ; une simple pile de serveurs d'applications peut se présenter comme suit.



En plus de créer les instances et d'installer les packages nécessaires, vous avez généralement besoin d'un moyen de distribuer les applications aux serveurs d'applications, de surveiller les performances de la pile, de gérer les autorisations et la sécurité, et ainsi de suite.

AWS OpsWorks Stacks fournit un moyen simple et flexible de créer et de gérer des piles et des applications.

Voici à quoi peut ressembler une pile de serveurs d'applications de base avec AWS OpsWorks Stacks. Il se compose d'un groupe de serveurs d'applications exécutés derrière un équilibreur de charge Elastic Load Balancing, avec un serveur de base de données Amazon RDS principal.



Bien que relativement simple, cette pile présente toutes les fonctionnalités clés de AWS OpsWorks Stacks. Voici comment la mettre en place.

Rubriques

- Piles
- Couches
- Recettes et LifeCycle événements
- instances
- Applications
- Personnalisation de votre pile
- Gestion des ressources

- Sécurité et autorisations
- Surveillance et journalisation
- CLI, SDK et modèles AWS CloudFormation
- AWS OpsWorks Stacks Fin de vie FAQs
- Migration de vos AWS OpsWorks Stacks applications vers AWS Systems Manager Application Manager
- Utilisation de l' AWS OpsWorks Stacks outil Detach in Place
- Débuter avec AWS OpsWorks Stacks
- AWS OpsWorks Meilleures pratiques de Stacks
- Piles
- Couches
- instances
- Applications
- Livres de recettes et recettes
- Gestion des ressources
- Balises
- Surveillance
- Sécurité et autorisations
- AWS OpsWorks Support Stacks pour Chef 12 Linux
- Support pour les versions précédentes de Chef dans AWS OpsWorks Stacks
- Utilisation de AWS OpsWorks Stacks avec d'autres services AWS
- Utilisation de la AWS OpsWorks CLI Stacks
- Guide de débogage et dépannage
- AWS OpsWorks CLI de l'agent Stacks
- AWS OpsWorks Référence du sac de données Stacks
- OpsWorks Changements d'agent

Piles

La pile est le composant principal de AWS OpsWorks Stacks. Il s'agit essentiellement d'un conteneur pour les ressources AWS (instances Amazon EC2 , instances de base de données Amazon RDS,

etc.) qui ont un objectif commun et doivent être gérées logiquement ensemble. La pile vous permet de gérer ces ressources en tant que groupe et définit également certains paramètres de configuration par défaut, tels que le système d'exploitation et la région AWS des instances. Si vous souhaitez isoler certains composants de la pile d'une interaction directe de l'utilisateur, vous pouvez exécuter la pile dans un VPC.

Couches

Vous définissez les composants de la pile en ajoutant une ou plusieurs couches. Une couche représente un ensemble d' EC2 instances Amazon qui répondent à un objectif particulier, comme le service d'applications ou l'hébergement d'un serveur de base de données.

Vous pouvez personnaliser ou étendre les couches en modifiant les configurations par défaut des packages, en ajoutant des recettes Chef pour exécuter des tâches telles que l'installation de packages supplémentaires, et plus encore.

Pour toutes les piles, AWS OpsWorks Stacks inclut des couches de service, qui représentent les services AWS suivants.

- Amazon Relational Database Service
- Elastic Load Balancing
- Amazon Elastic Container Service

Les couches vous donnent un contrôle complet sur les packages installés, sur leur configuration, sur les applications qui y sont déployées, et plus encore.

Recettes et LifeCycle événements

Les couches dépendent des recettes Chef pour gérer des tâches telles que l'installation de packages sur les instances, le déploiement d'applications, l'exécution de scripts, etc. L'une des principales fonctionnalités de AWS OpsWorks Stacks est un ensemble d'événements du cycle de vie (installation, configuration, déploiement, annulation du déploiement et arrêt) qui exécutent automatiquement un ensemble spécifique de recettes au moment opportun sur chaque instance.

Chaque couche peut avoir un ensemble de recettes attribuées à chaque événement du cycle de vie et qui gèrent une grande variété de tâches pour cet événement et cette couche. Par exemple, une fois qu'une instance appartenant à une couche de serveur Web a fini de démarrer, AWS OpsWorks Stacks effectue les opérations suivantes.

1. Exécute les recettes Setup de la couche, qui peuvent exécuter des tâches telles que l'installation et la configuration d'un serveur web.
2. Exécute les recettes Deploy de la couche, qui déploient les applications de la couche depuis un référentiel vers l'instance et exécutent les tâches connexes, telles que le redémarrage du service.
3. Exécute les recettes Configure sur chaque instance de la pile de telle sorte que chaque instance puisse ajuster sa configuration en fonction des besoins pour accueillir la nouvelle instance.

   Par exemple, sur une instance exécutant un équilibreur de charge, une recette Configure peut modifier la configuration de l'équilibreur de charge pour inclure la nouvelle instance.

Si une instance appartient à plusieurs couches, AWS OpsWorks Stacks exécute les recettes pour chaque couche afin que vous puissiez, par exemple, avoir une instance qui supporte un serveur d'applications PHP et un serveur de base de données MySQL.

Si vous avez implémenté des recettes, vous pouvez attribuer chaque recette à la couche et à l'événement appropriés et AWS OpsWorks Stacks les exécute automatiquement pour vous au moment opportun. Vous pouvez aussi exécuter les recettes manuellement, à tout moment.

instances

Une instance représente une ressource informatique unique, telle qu'une EC2 instance Amazon. Elle définit la configuration de base de la ressource, telle que le système d'exploitation et la taille. Les autres paramètres de configuration, tels que les adresses IP élastiques ou les volumes Amazon EBS, sont définis par les couches de l'instance. Les recettes de la couche terminent la configuration en effectuant des tâches telles que l'installation et la configuration des packages, ou le déploiement d'applications.

Vous pouvez utiliser AWS OpsWorks Stacks pour créer des instances et les ajouter à une couche. Lorsque vous démarrez l'instance, AWS OpsWorks Stacks lance une EC2 instance Amazon en utilisant les paramètres de configuration spécifiés par l'instance et sa couche. Une fois le démarrage de l' EC2 instance Amazon terminé, AWS OpsWorks Stacks installe un agent qui gère la communication entre l'instance et le service et exécute les recettes appropriées en réponse aux événements du cycle de vie.

AWS OpsWorks Stacks prend en charge les types d'instances suivants, qui se caractérisent par leur mode de démarrage et d'arrêt.

- Les instances 24/7 sont démarrées manuellement et exécutées jusqu'à ce que vous les arrêtiez.

- Les instances basées sur le temps sont exécutées par AWS OpsWorks Stacks selon un calendrier quotidien et hebdomadaire spécifié.

  Ils permettent à votre pile d'ajuster automatiquement le nombre d'instances afin d'accueillir les modèles d'utilisation prévisibles.
- Les instances basées sur la charge sont automatiquement démarrées et arrêtées par AWS OpsWorks Stacks, en fonction de mesures de charge spécifiées, telles que l'utilisation du processeur.

  Elles permettent à votre pile d'ajuster automatiquement le nombre d'instances pour accueillir les fluctuations du trafic entrant. Les instances à charge définie ne sont disponibles que pour les piles basées sur Linux.

AWS OpsWorks Stacks prend en charge l'autoréparation des instances. Si un agent arrête de communiquer avec le service, AWS OpsWorks Stacks arrête et redémarre automatiquement l'instance.

Vous pouvez également intégrer des ressources informatiques basées sur Linux dans une pile créée en dehors de Stacks. AWS OpsWorks

- EC2 Instances Amazon que vous avez créées directement à l'aide de la EC2 console, de la CLI ou de l'API Amazon.
- Instances locales s'exécutant sur votre propre matériel, y compris les instances s'exécutant dans les machines virtuelles.

Une fois que vous avez enregistré l'une de ces instances, elle devient une instance AWS OpsWorks Stacks et vous pouvez la gérer de la même manière que les instances que vous créez avec AWS OpsWorks Stacks.

Applications

Vous stockez les applications et les fichiers associés dans un référentiel, tel qu'un compartiment Amazon S3. Chaque application est représentée par une application, qui spécifie le type d'application et contient les informations qui sont nécessaires pour déployer l'application depuis le référentiel vers les instances, telles que l'URL du référentiel et le mot de passe. Lorsque vous déployez une application, AWS OpsWorks Stacks déclenche un événement Deploy, qui exécute les recettes de déploiement sur les instances de la pile.

Vous pouvez déployer les applications de la manière suivante :

- Automatiquement : lorsque vous démarrez des instances, AWS OpsWorks Stacks exécute automatiquement les recettes de déploiement de l'instance.
- Manuellement : si vous avez une nouvelle application ou si vous souhaitez mettre à jour une application existante, vous pouvez exécuter manuellement les recettes de déploiement des instances en ligne.

Vous demandez généralement à AWS OpsWorks Stacks d'exécuter les recettes de déploiement sur l'ensemble de la pile, ce qui permet aux instances des autres couches de modifier leur configuration de manière appropriée. Cependant, vous pouvez limiter le déploiement à un sous-ensemble d'instances si, par exemple, vous voulez tester une nouvelle application avant de la déployer sur chaque instance serveur d'applications.

Personnalisation de votre pile

AWS OpsWorks Stacks offre diverses façons de personnaliser les couches afin de répondre à vos besoins spécifiques:

- Vous pouvez modifier la façon dont AWS OpsWorks Stacks configure les packages en remplaçant les attributs qui représentent les différents paramètres de configuration, ou même en remplaçant les modèles utilisés pour créer les fichiers de configuration.
- Vous pouvez étendre une couche existante en fournissant vos propres recettes pour exécuter des tâches telles que l'exécution de scripts ou l'installation et la configuration de packages non standard.

Toutes les piles peuvent inclure une ou plusieurs couches, qui ne commencent qu'avec un ensemble minimal de recettes. Vous ajoutez des fonctionnalités à la couche en implémentant des recettes pour gérer des tâches telles que l'installation de packages, le déploiement d'applications, etc. Vous regroupez vos recettes personnalisées et les fichiers associés dans un ou plusieurs livres de recettes et vous stockez les livres de recettes dans un référentiel tel qu'Amazon S3 ou Git.

Vous pouvez exécuter des recettes manuellement, mais AWS OpsWorks Stacks vous permet également d'automatiser le processus en prenant en charge un ensemble de cinq événements du cycle de vie :

- L'événement Configuration se produit sur une nouvelle instance après la réussite de son démarrage.
- L'événement Configurer se produit sur toutes les instances de la pile lorsqu'une instance accède à l'état en ligne ou le quitte.
- L'événement Déployer se produit lorsque vous déployez une application.
- L'événement Annuler un déploiement se produit lorsque vous supprimez une application.
- L'événement Fermeture se produit lorsque vous arrêtez une instance.

Chaque couche peut avoir un nombre quelconque de recettes assignées à chaque événement. Lorsqu'un événement du cycle de vie se produit sur l'instance d'une couche, AWS OpsWorks Stacks exécute les recettes associées. Par exemple, lorsqu'un événement Deploy se produit sur une instance de serveur d'applications, AWS OpsWorks Stacks exécute les recettes de déploiement de la couche pour télécharger l'application ou effectuer des tâches connexes.

Gestion des ressources

Vous pouvez intégrer d'autres ressources AWS, comme les adresses IP Elastic, à votre pile. Vous pouvez utiliser la console ou l'API AWS OpsWorks Stacks pour enregistrer des ressources auprès d'une pile, associer des ressources enregistrées à des instances ou les détacher de celles-ci, et déplacer des ressources d'une instance à l'autre.

Sécurité et autorisations

AWS OpsWorks Stacks s'intègre à AWS Identity and Access Management (IAM) pour fournir des moyens robustes de contrôler la manière dont les utilisateurs accèdent à AWS OpsWorks Stacks, notamment les suivants :

- Comment les utilisateurs individuels peuvent interagir avec chaque pile, par exemple s'ils peuvent créer des ressources de pile telles que des couches et des instances, ou s'ils peuvent utiliser SSH ou RDP pour se connecter aux instances Amazon EC2 d'une pile.
- Comment AWS OpsWorks Stacks peut agir en votre nom pour interagir avec les ressources AWS telles que les EC2 instances Amazon.
- Comment les applications exécutées sur des instances AWS OpsWorks Stacks peuvent accéder aux ressources AWS telles que les compartiments Amazon S3.

- Comment gérer les clés publiques SSH et les mots de passe RDP des utilisateurs, et se connecter à une instance.

Surveillance et journalisation

AWS OpsWorks Stacks propose plusieurs fonctionnalités pour vous aider à surveiller votre stack et à résoudre les problèmes liés à votre stack et à toutes les recettes. Pour toutes les piles :

- AWS OpsWorks Stacks fournit un ensemble de CloudWatch métriques personnalisées pour les stacks Linux, qui sont résumées pour votre commodité sur la page de surveillance.

  AWS OpsWorks Stacks prend en charge les CloudWatch métriques standard pour les stacks Windows. Vous pouvez les surveiller à l'aide de la CloudWatch console.
- CloudTrail journaux, qui enregistrent les appels d'API effectués par AWS OpsWorks Stacks ou au nom de celui-ci depuis votre compte AWS.
- Un journal des événements, qui répertorie tous les événements de votre pile.
- Les journaux Chef qui détaillent ce qui s'est passé pour chaque événement de cycle de vie sur chaque instance, comme les recettes exécutées et les erreurs intervenues.

Les piles basées sur Linux peuvent également inclure une couche principale Ganglia, que vous pouvez utiliser pour collecter et afficher des données de surveillance détaillées pour les instances de votre pile.

CLI, SDK et modèles AWS CloudFormation

Outre la console, AWS OpsWorks Stacks prend également en charge une interface de ligne de commande (CLI) et plusieurs langues qui peuvent être utilisées SDKs pour effectuer n'importe quelle opération. Notez ces fonctionnalités :

- La CLI AWS OpsWorks Stacks fait partie de l'AWS CLI et peut être utilisée pour effectuer n'importe quelle opération depuis la ligne de commande.

  L'interface de ligne de commande AWS prend en charge plusieurs services AWS et peut être installée sur les systèmes Windows, Linux ou OS X.

- AWS OpsWorks Stacks est inclus dans les outils AWS pour Windows PowerShell et peut être utilisé pour effectuer n'importe quelle opération à partir d'une ligne de PowerShell commande Windows.
- Le SDK AWS OpsWorks Stacks est inclus dans l'AWS SDKs, qui peut être utilisé par des applications implémentées dans : Java JavaScript(basé sur un navigateur et Node.js), .NET, PHP,Python (boto) ou Ruby.

Vous pouvez également utiliser des AWS CloudFormation modèles pour approvisionner des piles. Pour quelques exemples, consultez AWS OpsWorks Snippets.

AWS OpsWorks Stacks Fin de vie FAQs

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible.

Rubriques

- Comment les clients existants seront-ils affectés par cette fin de vie ?
- Est-ce AWS OpsWorks Stacks accepter de nouveaux clients ?
- Où dois-je migrer mes piles existantes ?
- Comment puis-je conserver mes EC2 instances Amazon existantes après leur fin de vie ?
- La fin de vie aura-t-elle un impact sur Régions AWS tout le monde en même temps ?
- Quel est le niveau de support technique disponible AWS OpsWorks Stacks ?
- Y aura-t-il de nouvelles fonctionnalités pour AWS OpsWorks Stacks ?

Comment les clients existants seront-ils affectés par cette fin de vie ?

Les clients existants ne seront pas concernés jusqu'au 26 mai 2024, date de fin de vie de. AWS OpsWorks Stacks Après le 26 mai 2024, les clients ne pourront plus utiliser la OpsWorks console, l'API, la CLI et les CloudFormation ressources.

Est-ce AWS OpsWorks Stacks accepter de nouveaux clients ?

Non AWS OpsWorks Stacks n'accepte plus de nouveaux clients et seuls les clients existants sont en mesure de créer de nouvelles piles pour le moment.

Où dois-je migrer mes piles existantes ?

Nous recommandons AWS OpsWorks Stacks aux clients de migrer leurs charges de travail vers un AWS Systems Manager endroit où ils peuvent tirer parti des fonctionnalités suivantes :

- Versions de Modern Chef
- SSM Agent
- Application Load Balancers
- Fonctionnalités de mise à l'échelle améliorées via les groupes Auto Scaling
- Possibilité de définir les caractéristiques d'hôte souhaitées à l'aide de modèles de EC2 lancement
- Nouveaux types d'instances
- Nouveaux types de volumes EBS

Pour plus d'informations sur Systems Manager, consultez le guide de AWS Systems Manager l'utilisateur. Pour plus d'informations sur la migration vers AWS Systems Manager, voir Migration de vos AWS OpsWorks Stacks applications vers AWS Systems Manager Application Manager

Comment puis-je conserver mes EC2 instances Amazon existantes après leur fin de vie ?

Une fois la date de fin de vie atteinte, vos EC2 instances Amazon resteront dans votre compte, mais vous ne pourrez plus utiliser le service OpsWorks Stacks pour contrôler et gérer les instances.

Vous pouvez utiliser l'outil AWS OpsWorks Stacks Detach in Place pour détacher vos OpsWorks instances du service OpsWorks Stacks. Après le détachement, vous pouvez utiliser Amazon EC2 ou toute autre approche EC2 compatible pour configurer et gérer les instances. AWS Systems Manager Pour de plus amples informations, veuillez consulter Utilisation de l' AWS OpsWorks Stacks outil Detach in Place.

La fin de vie aura-t-elle un impact sur Régions AWS tout le monde en même temps ?

Oui. La OpsWorks console, l'API, la CLI et les CloudFormation ressources seront toutes interrompues Régions AWS simultanément le 26 mai 2024. Pour une liste des Régions AWS endroits où ces services AWS OpsWorks Stacks sont disponibles, consultez la liste des services AWS régionaux.

Quel est le niveau de support technique disponible AWS OpsWorks Stacks ?

AWS continuera à fournir le même niveau de support AWS OpsWorks Stacks que celui dont bénéficient les clients aujourd'hui jusqu'à la date de fin de vie. Si vous avez des questions ou des préoccupations, vous pouvez contacter l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Y aura-t-il de nouvelles fonctionnalités pour AWS OpsWorks Stacks ?

Non Le service étant sur le point d'atteindre sa fin de vie, nous ne publierons aucune nouvelle fonctionnalité. Cependant, nous continuerons à améliorer la sécurité et à gérer les EC2 instances Amazon comme prévu jusqu'à la date de fin de vie.

Migration de vos AWS OpsWorks Stacks applications vers AWS Systems Manager Application Manager

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible.

Vous pouvez désormais migrer vos AWS OpsWorks Stacks applications vers Application Manager, une fonctionnalité de AWS Systems Manager, à l'aide d'un script de migration. La migration de vos applications Stacks vers Systems Manager Application Manager vous permet d'utiliser des AWS fonctionnalités qui ne sont pas disponibles dans AWS OpsWorks Stacks, telles que les nouveaux

types d' EC2 instances Amazon tels que Graviton, les nouveaux volumes Amazon Elastic Block Store (EBS) tels que gp3, les nouveaux systèmes d'exploitation, les intégrations avec les groupes Auto Scaling et les équilibreurs de charge des applications.

Avec cette version, vous pouvez désormais surveiller et exécuter des opérations sur vos instances migrées à l'aide d'un nouvel onglet Instances disponible dans Systems Manager Application Manager. Vous pouvez utiliser l'onglet Instances pour afficher plusieurs AWS instances au même endroit. Cet onglet vous permet d'afficher des informations sur l'état de santé de l'instance et de résoudre les problèmes. Pour plus d'informations sur l'utilisation de l'onglet Instances, consultez la section Utilisation des instances de votre application dans le Guide de AWS Systems Manager l'utilisateur.

Rubriques

- Comment fonctionne le script
- Prérequis
- Limites
- Premiers pas
- FAQ
- Résolution des problèmes

Comment fonctionne le script

AWS OpsWorks fournit un script que vous pouvez exécuter pour migrer vos AWS OpsWorks Stacks applications vers Systems Manager Application Manager à l'aide d'un CloudFormation modèle. Le script obtient des informations sur une OpsWorks couche existante et, en fonction de la valeur du `--provision-application` paramètre du script, fournit un clone de votre application ou fournit un CloudFormation modèle de démarrage que vous pouvez modifier à l'aide de celui-ci AWS CloudFormation.

Prérequis

- Assurez-vous que le AWS CLI est installé et configuré. Pour plus d'informations sur l'installation du AWS CLI, voir Installation ou mise à jour de la dernière version du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

> **Note**
>
> Si vous ne souhaitez pas configurer le AWS CLI, vous pouvez également exécuter des commandes à l'aide de AWS CloudShell. Pour plus d'informations sur l'utilisation CloudShell, consultez la section Travailler avec AWS CloudShell dans le guide de AWS CloudShell l'utilisateur.

- Assurez-vous que la version 3.6 ou ultérieure de Python est installée ou qu'elle est fournie avec l'Amazon Machine Image (AMI).
- Assurez-vous que votre système d'exploitation est compatible. Vous pouvez télécharger et exécuter le script de migration sur les systèmes d'exploitation suivants.
  - Amazon Linux et Amazon Linux 2
  - Ubuntu 18,04 LTS, 20,04 LTS, 22,04 LTS
  - Red Hat Enterprise Linux 8
  - Windows Server 2019, Windows 10 Entreprise

> **Note**
>
> Windows Server 2022 n'est pas pris en charge.

Limites

La nouvelle OpsWorks architecture est différente de celle de AWS OpsWorks Stacks. Cette section décrit les limites connues de cette architecture.

Les éléments suivants ne sont pas pris en charge par la nouvelle OpsWorks architecture.

- Exécution de recettes Chef sur des instances Windows et CentOS
- Chef 11 couches et Berkshelf intégrés
- Attributs du chef et sacs de données
- Instances sur site
- Instances importées depuis EC2
- Aucune prise en charge pour l'installation d'une liste de packages de système d'exploitation spécifiée par l'utilisateur

- Les applications ne sont pas prises en charge ou migrées

Les éléments suivants sont pris en charge avec certaines restrictions.

- Le script de migration clone les informations des volumes EBS, mais exclut les points de montage et les données réelles contenues dans les volumes.
- Les instances dimensionnées basées sur le temps et sur la charge sont migrées, mais les règles de dimensionnement associées à ces instances ne sont pas migrées. Vous pouvez modifier le groupe Auto Scaling pour obtenir des résultats similaires.
- Les entités IAM définies dans la page Permissions de la pile dans la OpsWorks console ne sont ni créées ni générées.
- Le script de migration est uniquement capable de provisionner des applications monocouche dans Systems Manager. Par exemple, si vous exécutez le script deux fois pour deux couches d'une même pile, vous obtenez deux applications différentes dans Systems Manager.

Premiers pas

Le script de migration est un script Python que vous pouvez exécuter localement ou sur une EC2 instance. `stack_exporter.py` Avant d'exécuter le script, assurez-vous que tous les prérequis sont remplis. Pour en savoir plus sur les prérequis, consultezPrérequis.

Les étapes décrites dans les sections suivantes vous montrent comment migrer vos OpsWorks piles vers Systems Manager Application Manager.

Rubriques

- Étape 1 : préparer votre environnement pour exécuter le script
- Étape 2 : Téléchargez le script de migration
- Étape 3 : configurer votre environnement pour exécuter le script
- Étape 4 : Exécuter le script
- Étape 5 : provisionner une CloudFormation pile
- Étape 6 : passer en revue les ressources allouées
- Étape 7 : démarrer une instance
- Étape 8 : passer en revue l'instance
- Étape 9 : Surveillez et exécutez les opérations sur vos instances à l'aide de Systems Manager Application Manager

Étape 1 : préparer votre environnement pour exécuter le script

Préparez votre environnement en exécutant les commandes appropriées à votre système d'exploitation.

Rubriques

- Amazon Linux 2
- Amazon Linux
- Ubuntu 18.04, 20.04, 22.04
- Red Hat Enterprise Linux 8
- Windows Server 2019, Windows 10 Entreprise

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
```

Pour Python version 3.6, exécutez également :

```
python3 -m pip install pipenv==2022.4.8
```

Pour les versions 3.7 et ultérieures de Python, exécutez également :

```
python3 -m pip install pipenv
```

Ubuntu 18.04, 20.04, 22.04

```
sudo su
export PATH="${HOME}/.local/bin:$PATH"
apt-get update
apt install python3-pip
apt-get install git // if git is not installed
python3 -m pip install --user pipenv==2022.4.8
```

Red Hat Enterprise Linux 8

```
sudo su
sudo dnf install python3
PATH="$PATH:/usr/local/bin"
yum update
yum install git
python3 -m pip install pipenv==2022.4.8
```

Windows Server 2019, Windows 10 Entreprise

> **Note**
>
> Pour Windows Server 2019, installez Python version 3.6.1 ou ultérieure.

```
pip install pipenv
```

Si Git n'est pas déjà installé, téléchargez et installez Git.

Si vous utilisez Git comme source de livre de recettes, ajoutez votre serveur Git à un `known_hosts` fichier avant d'exécuter le script sous Windows. Vous pouvez l'utiliser PowerShell pour créer la fonction suivante.

```
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."
   }
}
```

Vous pouvez ensuite fournir votre serveur Git (par exemple, github.com, git-codecommit). *repository_region*.amazonaws.com) lorsque vous exécutez la fonction.

```
add_to_known_hosts "myGitServer"
```

Étape 2 : Téléchargez le script de migration

Téléchargez le fichier zip contenant le script de migration et tous les fichiers pertinents en exécutant la commande suivante.

```
aws s3api get-object \
    --bucket export-opsworks-stacks-bucket-prod-us-east-1 \
    --key export_opsworks_stacks_script.zip export_opsworks_stacks_script.zip
```

Si vous utilisez Linux, installez l'utilitaire de décompression à l'aide des commandes suivantes.

```
sudo apt-get install unzip
sudo yum install unzip
```

Décompressez les fichiers à l'aide de la commande appropriée à votre système d'exploitation.

Pour Linux, utilisez la commande suivante.

```
unzip export_opsworks_stacks_script.zip
```

Pour Windows, utilisez la `Expand-Archive` commande dans PowerShell.

```
Expand-Archive -LiteralPath PathToZipFile -DestinationPath PathToDestination
```

Une fois le fichier décompressé, les répertoires et fichiers suivants sont disponibles.

- Lisez-moi .md
- LICENCE
- NOTICE
- requirements.txt
- modèles/
  - OpsWorksCFNTemplate.yaml
  - Monter EBSVolumes .yaml
- opsworks/
- formation de nuages/
- onglet_instances/
- cfn_stack_deployer.py
- s3.py
- stack_exporter_context.py
- stack_exporter.py

Étape 3 : configurer votre environnement pour exécuter le script

Configurez votre environnement pour exécuter le script à l'aide de la commande suivante.

```
pipenv install -r requirements.txt
pipenv shell
```

> **Note**
>
> À l'heure actuelle, le script ne peut fournir que des applications monocouches dans le gestionnaire d'applications. Par exemple, si vous exécutez le script deux fois pour deux couches d'une même pile, le script crée deux applications différentes dans Application Manager.

Après avoir configuré votre environnement, passez en revue les paramètres du script. Vous pouvez consulter les options disponibles pour le script de migration en exécutant la `python3 stack_exporter.py --help` commande.

Paramètre	Description	Obligatoire	Type	Valeur par défaut
`--layer-id`	Exporte un CloudFormation modèle pour cet ID de OpsWorks couche.	Oui	chaîne	
`--region`	La AWS région de la OpsWorks pile. Si votre région de OpsWorks pile et votre région de point de terminaison d'API sont différent es, utilisez la région de pile. Il s'agit de la même région que les autres ressources de votre OpsWorks pile (par exemple, les EC2 instances et les sous-rése aux).	Non	chaîne	us-east-1
`--provision-application`	Par défaut, le script approvisi onne l'application exportée par le CloudFormation modèle. Transmettez ce paramètre au script avec la valeur FALSE pour ignorer le provisionnement du CloudFormation modèle.	Non	Booléen	`TRUE`
`--launch-template`	Ce paramètre définit s'il faut utiliser un modèle de lancement existant ou créer un nouveau modèle de lancement. Vous pouvez créer un nouveau modèle de lancement qui utilise les propriétés d'instance recommand ées ou qui utilise des propriétés	Non	chaîne	`RECOMMENDED`

Paramètre	Description	Obligatoire	Type	Valeur par défaut
	d'instance correspondant à une instance en ligne. Les valeurs valides sont les suivantes : • `RECOMMENDED` - Utilise les caractéristiques d'instance de la dernière AMI pour le système d'exploitation de la OpsWorks pile et une taille d'instance c5.large. • `MATCH_LAST_INSTANCE` - Utilise les dernières caractéristiques des instances en ligne disponibles. • *`LaunchTemplateID`* / [*`LaunchTemplateVersion`*] - Utilise un modèle de lancement existant. Vous pouvez éventuellement fournir une version du modèle. Si vous ne fournissez pas de version de modèle, le script utilise la version par défaut.			

Paramètre	Description	Obligatoire	Type	Valeur par défaut
`--system-updates`	Définit s'il faut effectuer des mises à jour du noyau et du package lors du démarrage de l'instance. Les valeurs valides sont les suivantes : • `ALL_UPDATES` - Effectue les mises à jour du système pour le noyau et les packages au démarrage de l'instance. • `NO_UPDATES` - N'effectue pas de mises à jour du système au démarrage de l'instance. • `MATCH_LAYER_SETTINGS` - Utilise la `InstallUpdatesOnBoot` propriété de la OpsWorks couche ou de l'instance pour déterminer s'il faut installer les mises à jour du système.	Non	chaîne	`ALL_UPDATES`
`--http-username`	Nom du `SecureString` paramètre Systems Manager qui stocke le nom d'utilisateur utilisé pour s'authentifier auprès de l'archive HTTP contenant les livres de recettes personnalisés.	Non	chaîne	
`--http-password`	Nom du `SecureString` paramètre Systems Manager qui stocke le mot de passe utilisé pour s'authentifier auprès de l'archive HTTP contenant les livres de recettes personnalisés.	Non	chaîne	

Paramètre	Description	Obligatoire	Type	Valeur par défaut
`--repo-private-key`	Nom du `SecureString` paramètre Systems Manager qui stocke la clé SSH utilisée pour s'authentifier auprès du référentiel contenant les livres de recettes personnalisés. Si le dépôt est activé GitHub, vous devez générer une nouvelle clé `Ed25519` SSH. Si vous ne générez pas de nouvelle clé `Ed25519` SSH, la connexion au GitHub dépôt échoue.	Non	chaîne	
`--lb-type`	Le type d'équilibreur de charge, le cas échéant, à créer lors de la migration de votre équilibreur de charge existant. Les valeurs valides sont les suivantes : • `ALB`(Application Load Balancer) • `Classis`(Classic Load Balancer) • `None`(si vous ne souhaitez pas créer d'équilibreur de charge)	Non	chaîne	`ALB`

Paramètre	Description	Obligatoire	Type	Valeur par défaut
`--lb-access-logs-path`	Chemin d'accès à un compartiment S3 existant et préfixe pour stocker les journaux d'accès à l'équilibreur de charge. Le compartiment S3 et l'équilibreur de charge doivent se trouver dans la même région. Si vous ne fournissez aucune valeur et que la valeur du `--lb-type` paramètre est définie sur`None`, le script crée un nouveau compartiment et un nouveau préfixe S3. Assurez-vous qu'il existe une politique de compartiment appropriée pour ce préfixe.	Non	chaîne	
`--enable-instance-protection`	S'il est défini sur`TRUE`, le script crée une politique de résiliation personnalisée (fonction Lambda) pour votre groupe Auto Scaling. EC2les instances dotées d'une `protected_instance` balise sont protégées contre les événements de scale-in. Ajoutez une `protected_instance` balise à chaque EC2 instance que vous souhaitez protéger contre les événements d'extension.	Non	Booléen	`FALSE`

Paramètre	Description	Obligatoire	Type	Valeur par défaut
`--command-logs-bucket`	Le nom d'un compartiment S3 existant pour stocker les `MountEBSVolumes` journaux `AWS ApplyChefRecipe` et. Si vous ne fournissez aucune valeur, le script crée un nouveau compartiment S3.	Non	chaîne	`aws-opsworks-application-manager-logs-` *account-id*
`--custom-json-bucket`	Le nom d'un compartiment S3 existant pour stocker le JSON personnalisé. Si vous ne fournissez aucune valeur, le script crée un nouveau compartiment S3.	Non	chaîne	`aws-apply-chef-application-manager-transition-data-`*account-id*

Remarques :

- Si vous utilisez un GitHub dépôt privé, vous devez créer une nouvelle clé d'`Ed25519`hôte pour SSH. Cela est dû au fait que les clés prises en charge par SSH ont été GitHub modifiées et le protocole Git non chiffré a été supprimé. Pour plus d'informations sur la clé `Ed25519` d'hôte, consultez le billet de GitHub blog Améliorer la sécurité du protocole Git sur GitHub. Après avoir généré une nouvelle clé `Ed25519` d'hôte, créez un `SecureString` paramètre Systems Manager pour la clé SSH et utilisez le nom du `SecureString` paramètre comme valeur du `--repo-private-key` paramètre. Pour plus d'informations sur la création d'un `SecureString` paramètre Systems Manager, voir Create a SecureString parameter (AWS CLI) ou Create a Systems Manager parameter (console) dans le Guide de AWS Systems Manager l'utilisateur.
- Les `--http-username` `--repo-private-key` paramètres `--http-password` et font référence au nom d'un `SecureString` paramètre de Systems Manager. Le script de migration utilise ces paramètres lorsque vous exécutez le `AWS-ApplyChefRecipes` document.
- Le `--http-username` paramètre nécessite également que vous spécifiiez une valeur pour le `--http-password` paramètre.
- Le `--http-password` paramètre nécessite également que vous spécifiiez une valeur pour le `--http-username` paramètre.

- Ne définissez pas de valeurs à la fois pour `--http-password` et`--repo-private-key`. Indiquez soit le nom du `SecureString` paramètre Systems Manager correspondant à une clé SSH (`--repo-private-key`), soit le nom d'utilisateur (`--http-username`) et le mot de passe (`--http-password`) du référentiel.

Étape 4 : Exécuter le script

Lorsque vous l'exécutez`python3 stack_exporter.py`, vous pouvez soit provisionner l'application, soit créer un modèle de démarrage en définissant la valeur du `--provision-application` paramètre sur`FALSE`.

Exemple 1 : mise en service d'une application Systems Manager Application Manager

La commande suivante obtient des informations sur une OpsWorks couche existante et provisionne une application à l'aide de la nouvelle OpsWorks architecture, ce qui permet d'obtenir un résultat similaire à la version de Chef configurée pour la pile. Le script fournit toutes les ressources requises, telles que les groupes Auto Scaling CloudFormation, en utilisant, puis enregistre l'application dans Systems Manager Application Manager.

Remplacez *`stack-region`* et *`layer-id`* par les valeurs de votre OpsWorks pile et de votre couche.

```
python3 stack_exporter.py \
     --layer-id layer-id \
     --region stack-region
```

Exemple 2 : générer un modèle

La commande suivante permet d'obtenir des informations sur une OpsWorks couche existante et de générer un CloudFormation modèle. Le modèle, s'il est provisionné, obtient un résultat similaire à celui obtenu avec Chef 14. Dans cet exemple, aucune ressource n'est provisionnée, car le `--provision-application` paramètre est défini sur. `FALSE`

Remplacez *`stack-region`* et *`layer-id`* par les valeurs de votre OpsWorks pile et de votre couche.

```
python3 stack_exporter.py \
    --layer-id layer-id \
    --region stack-region \
    --provision-application FALSE
```

Après avoir exécuté la commande, vous pouvez consulter le modèle dans la bibliothèque de modèles Application Manager dans Systems Manager, et vous pouvez également provisionner le modèle. Pour plus d'informations sur l'affichage de la bibliothèque de modèles, voir Utilisation de la bibliothèque de modèles dans le Guide de AWS Systems Manager l'utilisateur.

Étape 5 : provisionner une CloudFormation pile

> **Note**
>
> Vous ne devez effectuer cette étape que si vous définissez le `--provision-application` paramètre du script sur`FALSE`.

Lorsque vous spécifiez le `--provision-application` paramètre avec une valeur de`FALSE`, la sortie du script fournit le nom et l'URL du CloudFormation modèle. Ce modèle représente une proposition de remplacement de votre OpsWorks pile et de votre couche existantes.

Vous pouvez configurer le modèle à l'aide de la bibliothèque de modèles d'Application Manager (recommandée) ou en utilisant CloudFormation. Pour plus d'informations sur l'utilisation de la bibliothèque de modèles, voir Utilisation de la bibliothèque de modèles dans le Guide de AWS Systems Manager l'utilisateur.

Étape 6 : passer en revue les ressources allouées

Vous êtes maintenant prêt à passer en revue les ressources allouées.

1. Passez en revue les ressources de la pile provisionnée à l'aide de la AWS CloudFormation console.
   a. Ouvrez la AWS CloudFormation console dans https://console.aws.amazon.com/cloudformation et choisissez Stacks.
   b. Sur la page Stacks, choisissez la pile, puis cliquez sur l'onglet Ressources.
   c. Dans l'onglet Ressources, passez en revue les ressources répertoriées pour votre pile. La liste des ressources inclut un groupe EC2 Auto Scaling, que vous pouvez consulter dans la console Auto Scaling, ou AWS CLI.
2. Passez en revue les ressources de l'application à l'aide de Systems Manager Application Manager.

a. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.

b. Dans le volet de navigation, choisissez Application Manager.

c. Dans la section Applications, choisissez l'application personnalisée. Le gestionnaire d'applications ouvre l'onglet Vue d'ensemble.

d. Sélectionnez l'onglet Ressources. L'onglet Ressources affiche toutes les ressources qui ont été migrées pour votre OpsWorks pile et votre couche. Le nom de l'application inclut le nom de la OpsWorks pile et est formaté sous la forme *app* - *stack-name* - *suffix* où *suffix* représente les six premiers caractères de l'ID de pile. Pour plus d'informations sur l'affichage des ressources dans le Gestionnaire d'applications, consultez la section Affichage des ressources des applications dans le Guide de AWS Systems Manager l'utilisateur.

Étape 7 : démarrer une instance

Une fois que vous avez provisionné une instance, vous êtes prêt à la tester. À ce stade, aucune instance n'est en cours d'exécution.

Pour mettre vos instances en ligne, ajustez les `Min` `Desired capacity` valeurs`Max`, et du groupe Auto Scaling selon un nombre adapté à votre application. Dans un premier temps, vous souhaiterez peut-être définir ces valeurs sur 1, pour mettre une seule instance en ligne et vérifier que l'instance exécute toutes les actions attendues, y compris l'exécution de vos recettes Chef personnalisées.

Étape 8 : passer en revue l'instance

Après avoir démarré une instance, vérifiez qu'elle fonctionne comme prévu.

1. Passez en revue le Chef `startup` et `terminate` les journaux situés dans le compartiment S3 spécifié par le `--command-logs-bucket` paramètre du script. Par défaut, les journaux sont stockés dans un compartiment portant le nom`aws-opsworks-application-manager-logs-`*`account-id`*.

   a. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

   b. Choisissez le compartiment contenant vos journaux.

   c. Accédez au `ApplyChefRecipes` préfixe pour afficher vos journaux.

2. Vérifiez la connectivité et l'état de santé de l'Application Load Balancer.

Procédez comme suit pour consulter les journaux d'accès à votre équilibreur de charge. Vous pouvez spécifier le compartiment S3 dans lequel vous souhaitez stocker les journaux d'accès à l'équilibreur de charge à l'aide du `--lb-access-logs-path` paramètre du script.

a. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

b. Choisissez votre compartiment S3, puis accédez au préfixe contenant vos journaux.

3. Vérifiez que l'instance passe tous les tests de santé d'Auto Scaling et d'Application Load Balancer (si vous en avez configuré un).

   Vous pouvez consulter les informations relatives à l'état de santé d'Auto Scaling dans le nouvel onglet Instances.

   a. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.

   b. Dans le volet de navigation, choisissez Application Manager.

   c. Dans la section Applications, sélectionnez Applications personnalisées.

   d. Choisissez l'application dans la liste. Le gestionnaire d'applications ouvre l'onglet Vue d'ensemble.

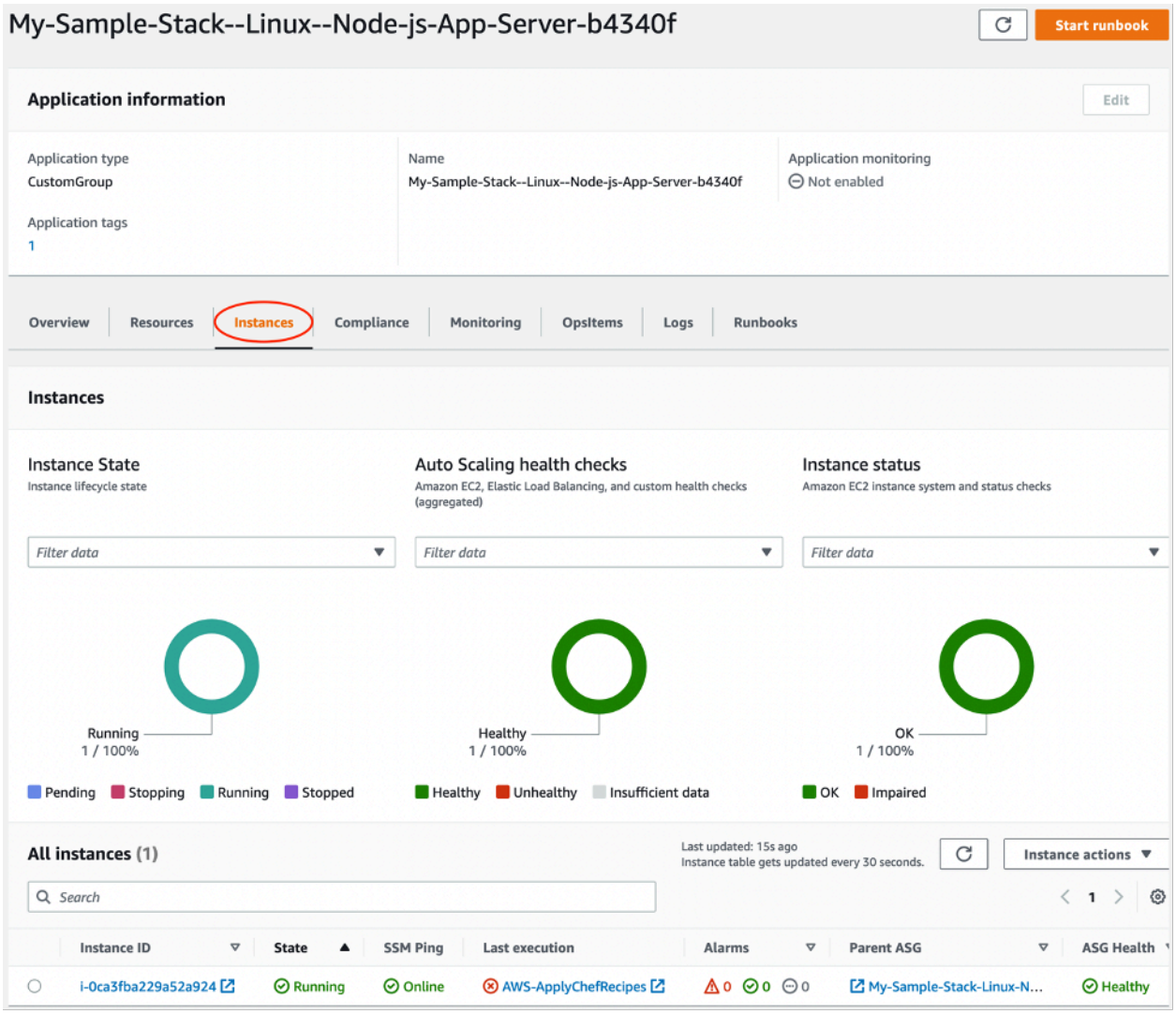
   e. Choisissez l'onglet Instances pour afficher les informations relatives à l'état de santé d'Auto Scaling.

Après avoir vérifié que les recettes Chef s'exécutent correctement, vous pouvez réduire la capacité du groupe Auto Scaling pour mettre fin à l'instance. Si vous avez des recettes de terminaison personnalisées, vérifiez qu'elles fonctionnent comme prévu.

Étape 9 : Surveillez et exécutez les opérations sur vos instances à l'aide de Systems Manager Application Manager

Vous pouvez désormais surveiller et exécuter des opérations sur vos instances à l'aide d'un nouvel onglet Instances sur la page Gestionnaire d'applications. Pour plus d'informations sur l'utilisation de l'onglet Instances, consultez la section Utilisation des instances de votre application dans le Guide de AWS Systems Manager l'utilisateur.

Vous pouvez utiliser l'onglet Instances pour afficher plusieurs AWS instances au même endroit. Cet onglet vous permet d'afficher des informations sur l'état de santé de l'instance et de résoudre les problèmes.



Procédez comme suit pour afficher l'onglet Instances.

1. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.
2. Dans le volet de navigation, choisissez Application Manager.
3. Dans la section Applications, sélectionnez Applications personnalisées.
4. Choisissez l'application dans la liste. Le gestionnaire d'applications ouvre l'onglet Vue d'ensemble.
5. Choisissez l'onglet Instances pour afficher les informations relatives à l'état et à l'état de EC2 santé de votre instance.

FAQ

Vous trouverez ci-dessous FAQs des réponses aux questions les plus fréquemment posées.

Rubriques

- Quelles AWS OpsWorks Stacks versions puis-je migrer ?
- Quelles versions de Chef peuvent utiliser mes instances migrées ?
- Quels types de référentiels puis-je migrer ?
- Puis-je continuer à utiliser un dépôt Git privé ?
- Quelles clés SSH puis-je utiliser pour accéder à mes instances ?
- Pourquoi mes instances évoluent-elles automatiquement vers l'avant et vers l'extérieur ?
- Puis-je désactiver Auto Scaling ?
- Puis-je effectuer des mises à jour du noyau et des packages sur les EC2 instances lancées ?
- Pourquoi les volumes EBS de mes instances ne contiennent-ils aucune donnée ?
- Pourquoi les volumes EBS décrits dans mon modèle de lancement ne sont-ils pas montés ?
- Où puis-je trouver les journaux de volume des recettes Chef et Mount EBS ?
- Où puis-je trouver le journal de débogage du script de migration ?
- Le script de migration prend-il en charge la gestion des versions des CloudFormation modèles ?
- Puis-je migrer plusieurs couches ?
- Comment créer un SecureString paramètre ?
- Comment puis-je protéger les instances du nouveau groupe Auto Scaling contre les événements de résiliation ?
- Quels équilibreurs de charge sont disponibles avec le script de migration ?
- Les recettes de configuration de livres de recettes personnalisées sont-elles migrées ?
- Puis-je exécuter des recettes de déploiement et de dédéploiement sur mes instances nouvellement créées ?
- Puis-je modifier les sous-réseaux couverts par mon groupe Auto Scaling ?

Quelles AWS OpsWorks Stacks versions puis-je migrer ?

Vous ne pouvez migrer que les stacks Chef 11.10 et Chef 12, Amazon Linux, Amazon Linux 2, Ubuntu et Red Hat Enterprise Linux 7.

Quelles versions de Chef peuvent utiliser mes instances migrées ?

Les instances migrées peuvent utiliser les versions 11 à 14 de Chef.

> **Note**
>
> La migration vers Windows Stack n'est pas prise en charge.

Quels types de référentiels puis-je migrer ?

Vous pouvez migrer les types de référentiels S3, Git et HTTP.

Puis-je continuer à utiliser un dépôt Git privé ?

Oui, vous pouvez continuer à utiliser un dépôt Git privé.

Si vous utilisez un GitHub dépôt privé, vous devez créer une nouvelle clé d'`Ed25519`hôte pour SSH. Cela est dû au fait que les clés prises en charge par SSH ont été GitHub modifiées et le protocole Git non chiffré a été supprimé. Pour plus d'informations sur la clé `Ed25519` d'hôte, consultez le billet de GitHub blog Améliorer la sécurité du protocole Git sur GitHub. Après avoir généré une nouvelle clé `Ed25519` d'hôte, créez un `SecureString` paramètre Systems Manager pour cette clé SSH et utilisez le nom du paramètre comme valeur du `--repo-private-key` paramètre. Pour plus d'informations sur la création d'un `SecureString` paramètre Systems Manager, voir Create a SecureString parameter (AWS CLI) dans le Guide de AWS Systems Manager l'utilisateur.

Pour tout autre type de dépôt Git, créez un `SecureString` paramètre Systems Manager pour cette clé SSH et utilisez le nom du paramètre comme valeur du `--repo-private-key` paramètre du script.

Quelles clés SSH puis-je utiliser pour accéder à mes instances ?

Lorsque vous exécutez le script, celui-ci migre les clés SSH et les instances configurées dans la pile. Vous pouvez utiliser les clés SSH pour accéder à votre instance. Si des clés SSH sont fournies pour la pile et l'instance, le script utilise les clés de la pile. Si vous ne savez pas quelles clés SSH utiliser, consultez les instances dans la EC2 console (https://console.aws.amazon.com/ec2/). La page Détails de la EC2 console affiche les clés SSH de votre instance.

Pourquoi mes instances évoluent-elles automatiquement vers l'avant et vers l'extérieur ?

Auto Scaling redimensionne les instances en fonction des règles de dimensionnement du groupe Auto Scaling. Vous pouvez définir les valeurs de capacité minimale, maximale et souhaitée pour votre groupe. Le groupe Auto Scaling adapte automatiquement votre capacité en conséquence lorsque vous mettez à jour ces valeurs.

Puis-je désactiver Auto Scaling ?

Vous pouvez désactiver Auto Scaling en définissant les valeurs de capacité minimale, maximale et souhaitée du groupe Auto Scaling sur le même nombre. Par exemple, si vous souhaitez toujours avoir dix instances, définissez les valeurs de capacité minimale, maximale et souhaitée sur dix.

Puis-je effectuer des mises à jour du noyau et des packages sur les EC2 instances lancées ?

Par défaut, les mises à jour du noyau et des packages ont lieu au démarrage de l' EC2 instance. Procédez comme suit pour effectuer des mises à jour du noyau ou du package sur une EC2 instance lancée. Par exemple, vous souhaiterez peut-être appliquer des mises à jour après avoir exécuté des recettes de déploiement ou de configuration.

1. Connectez-vous à votre EC2 instance.
2. Créez la `perform_upgrade` fonction suivante et exécutez-la sur votre instance.

```
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. Après les mises à jour du noyau et du package, vous devrez peut-être redémarrer votre EC2 instance. Pour vérifier si un redémarrage est nécessaire, créez la `reboot_if_required` fonction suivante et exécutez-la sur votre EC2 instance.

```
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. Si les `reboot_if_required` résultats sont affichés dans un `reboot is required` message, redémarrez l' EC2 instance. Si vous recevez un `reboot is not required` message, il n'est pas nécessaire de redémarrer l' EC2 instance.

Pourquoi les volumes EBS de mes instances ne contiennent-ils aucune donnée ?

Lorsque vous exécutez le script, celui-ci migre la configuration des volumes EBS, créant ainsi une architecture de remplacement pour votre OpsWorks pile et votre couche. Le script ne migre pas les instances réelles ni les données qu'elles contiennent. Le script migre uniquement la configuration des volumes EBS au niveau de la couche et attache les volumes EBS vides aux instances lancées. EC2

Procédez comme suit pour extraire les données des volumes EBS de vos instances précédentes.

1. Prenez un instantané des volumes EBS de vos instances précédentes. Pour plus d'informations sur la création d'un instantané, consultez la section Créer un instantané Amazon EBS dans le guide de l' EC2 utilisateur Amazon.

2. Créez un volume à partir de votre instantané. Pour plus d'informations sur la création d'un volume à partir d'un instantané, consultez la section Créer un volume à partir d'un instantané dans le guide de EC2 l'utilisateur Amazon.
3. Attachez le volume que vous avez créé aux instances. Pour plus d'informations sur l'attachement de volumes, consultez la section Attacher un volume Amazon EBS à une instance dans le guide de l' EC2 utilisateur Amazon.

Pourquoi les volumes EBS décrits dans mon modèle de lancement ne sont-ils pas montés ?

Si vous fournissez un ID de modèle de lancement pour le `--launch-template` paramètre avec les volumes EBS, le script attache les volumes EBS, mais ne les monte pas. Vous pouvez monter les volumes EBS attachés en exécutant le `MountEBSVolumes` RunCommand document créé par le script pour l' EC2 instance lancée.

Si vous ne définissez aucun `--launch-template` paramètre, le script crée un modèle, et lorsque le groupe Auto Scaling lance une nouvelle EC2 instance, le groupe Auto Scaling attache automatiquement les volumes EBS, puis exécute la `SetupAutomation` commande pour monter les volumes attachés aux points de montage configurés dans les paramètres de couche.

Où puis-je trouver les journaux de volume des recettes Chef et Mount EBS ?

OpsWorks fournit les journaux dans un compartiment S3 que vous pouvez spécifier en fournissant une valeur pour le `--command-logs-bucket` paramètre. Le nom du compartiment S3 par défaut est au format :`aws-opsworks-stacks-application-manager-logs-`*`account-id`*. Les journaux de recettes du chef sont stockés dans le `ApplyChefRecipes` préfixe. Les journaux du volume Mount EBS sont stockés dans le `MountEBSVolumes` préfixe. Toutes les couches migrées à partir d'une pile fournissent des journaux vers le même compartiment S3.

> **Note**
>
> - La configuration du cycle de vie du compartiment S3 inclut une règle permettant de supprimer les journaux après 30 jours. Si vous souhaitez conserver les journaux pendant plus de 30 jours, vous devez mettre à jour la règle dans la configuration du cycle de vie du compartiment S3.

- Actuellement, il enregistre OpsWorks uniquement le chef `setup` et les `terminate` recettes.

Où puis-je trouver le journal de débogage du script de migration ?

Le script place les journaux de débogage dans un compartiment nomméaws-opsworks-stacks-transition-logs-*account-id*. Les journaux de débogage se trouvent dans le `migration_script` dossier du compartiment S3, sous les dossiers qui correspondent au nom de la couche que vous avez migrée.

Le script de migration prend-il en charge la gestion des versions des CloudFormation modèles ?

Le script génère des documents Systems Manager de type CloudFormation qui remplacent la couche ou la pile que vous souhaitez migrer. La réexécution du script, même avec les mêmes paramètres, permet d'exporter une nouvelle version du modèle de couche précédemment exporté. Les versions du modèle sont stockées dans le même compartiment S3 que les journaux de script.

Puis-je migrer plusieurs couches ?

Le `--layer-id` paramètre du script est transmis dans une seule couche. Pour migrer plusieurs couches, réexécutez le script et transmettez-en un autre`--layer-id`.

Les couches faisant partie d'une même OpsWorks pile sont répertoriées sous la même application dans Application Manager.

1. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.
2. Dans le volet de navigation, choisissez Application Manager.
3. Dans la section Applications, sélectionnez Applications personnalisées.
4. Choisissez votre application. Le nom de l'application commence parapp-*stack-name*-*first-six-characters-stack-id*.
5. L'élément de niveau supérieur, commençant par app, montre tous les composants qui correspondent à votre OpsWorks pile. Cela inclut les composants correspondant à votre OpsWorks couche.

6. Choisissez le composant correspondant à la couche pour afficher les ressources de la couche. Les composants représentant les OpsWorks couches sont également visibles dans la section Applications personnalisées en tant qu'applications individuelles.

Comment créer un **`SecureString`** paramètre ?

Vous pouvez utiliser Systems Manager pour créer un `SecureString` paramètre. Pour plus d'informations sur la création d'un `SecureString` paramètre Systems Manager, voir Create a SecureString parameter (AWS CLI) ou Create a Systems Manager parameter (console) dans le Guide de AWS Systems Manager l'utilisateur.

Vous devez fournir un `SecureString` paramètre comme valeur pour les `--repo-private-key` paramètres `--http-username--http-password`, ou.

Comment puis-je protéger les instances du nouveau groupe Auto Scaling contre les événements de résiliation ?

Vous pouvez protéger les instances en définissant le `--enable-instance-protection` paramètre sur `TRUE` et en ajoutant une clé de `protected_instance` balise à chaque EC2 instance que vous souhaitez protéger contre les événements de résiliation. Lorsque vous définissez le `--enable-instance-protection` paramètre sur `TRUE` et ajoutez une clé de `protected_instance` balise, le script ajoute une politique de résiliation personnalisée à votre nouveau groupe Auto Scaling et suspend le `ReplaceUnhealthy` processus. Les instances dotées de la clé de `protected_instance` balise sont protégées contre les événements de terminaison suivants :

- Échelle des événements
- Actualisation d'instance
- Rééquilibrage
- Durée de vie maximale de l'instance
- Autoriser la résiliation de l'instance de référencement
- Résiliation et remplacement d'instances défectueuses

> **Note**
>
> Vous devez définir la clé de `protected_instance` balise sur les instances que vous souhaitez protéger. La clé du tag distingue les majuscules et minuscules. Toute instance dotée de cette clé de balise est protégée quelle que soit la valeur de la balise.
> Pour réduire la durée d'exécution de la politique de résiliation personnalisée, vous pouvez augmenter le nombre d'instances par défaut utilisées par la fonction Lambda pour filtrer les instances protégées en mettant à jour la valeur de la variable de code de `default_sample_size` fonction. La valeur par défaut est 15. Si vous augmentez le`default_sample_size`, vous devrez peut-être augmenter la mémoire allouée à la fonction Lambda, ce qui augmentera le coût de votre fonction Lambda. Pour plus d'informations sur la tarification AWS Lambda , consultez Tarification AWS Lambda.

Quels équilibreurs de charge sont disponibles avec le script de migration ?

Le script propose trois options d'équilibreur de charge.

- (Recommandé) Créez un nouvel Application Load Balancer. Par défaut, le script crée un nouvel Application Load Balancer. Vous pouvez également définir le `--lb-type` paramètre surALB. Pour plus d'informations sur les équilibreurs de charge d'application, voir Qu'est-ce qu'un équilibreur de charge d'application ? dans le guide de l'utilisateur d'Elastic Load Balancing.
- Si un Application Load Balancer n'est pas une option, créez un Classic Load Balancer en définissant le paramètre `--lb-type` sur. `Classic` Si vous sélectionnez cette option, votre Classic Load Balancer existant attaché à votre OpsWorks couche est séparé de votre application. Pour plus d'informations sur les équilibreurs de charge d'application, voir Qu'est-ce qu'un Classic Load Balancer ? dans le guide de l'utilisateur d'Elastic Load Balancing : Classic Load Balancers.
- Vous pouvez associer un équilibreur de charge existant en définissant le `--lb-type` paramètre surNone.

> **Important**
>
> Nous vous recommandons de créer de nouveaux équilibreurs de charge Elastic Load Balancing pour vos couches AWS OpsWorks Stacks. Si vous choisissez d'utiliser un équilibreur de charge Elastic Load Balancing existant, vous devez d'abord vérifier qu'il n'est pas utilisé à d'autres fins et qu'aucune instance n'est attachée. Une fois que l'équilibreur de charge est attaché à la couche, il OpsWorks supprime toutes les instances existantes

et configure l'équilibreur de charge pour qu'il gère uniquement les instances de la couche. Bien qu'il soit techniquement possible d'utiliser la console ou l'API Elastic Load Balancing pour modifier la configuration d'un équilibreur de charge après l'avoir attaché à une couche, vous ne devez pas le faire ; les modifications ne seront pas permanentes.

Pour associer votre équilibreur de charge de OpsWorks couche existant à votre groupe Auto Scaling

1. Exécutez le script de migration avec le `--lb-type` paramètre défini sur`None`. Lorsque la valeur est définie sur`None`, le script ne clone ni ne crée d'équilibreur de charge.
2. Une fois que le script a déployé la CloudFormation pile, mettez à jour les groupes `Min` `Max` et les `Desired capacity` valeurs d'Auto Scaling, puis testez votre application.
3. Choisissez `Link to the template` affiché dans la sortie du script. Si vous avez fermé votre terminal, procédez comme suit pour accéder au modèle.
   a. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.
   b. Dans le volet de navigation, choisissez Application Manager.
   c. Choisissez des CloudFormation piles, puis choisissez Bibliothèque de modèles.
   d. Choisissez Owned by me et localisez votre modèle.
4. Dans le CloudFormation modèle, choisissez Modifier dans le menu Actions.
5. Mettez à jour la `LabelBalancerNames` propriété dans la section des `ApplicationAsg` ressources du CloudFormation modèle.

```
ApplicationAsg:
   DependsOn: CustomTerminationLambdaPermission
   Properties:
   #(other properties in ApplicationAsg to remain unchanged)
      LoadBalancerNames:
        - load-balancer-name
      HealthCheckType: ELB
```

6. Si vous souhaitez que la vérification de l'état de vos instances de groupe Auto Scaling utilise également celle de l'équilibreur de charge, supprimez la section ci-dessous `HealthCheckType` et entrez`ELB`. Si vous n'avez besoin que EC2 de bilans de santé, il n'est pas nécessaire de modifier le modèle.

7. Enregistrez vos modifications. L'enregistrement crée une nouvelle version par défaut du modèle. Si c'est la première fois que vous exécutez le script pour la couche et que vous enregistrez des modifications dans la console, la version la plus récente est 2.
8. Dans Actions, sélectionnez Provision stack.
9. Confirmez que vous souhaitez utiliser la version par défaut du modèle. Assurez-vous que l'option Sélectionner une pile existante est sélectionnée et choisissez la CloudFormation pile à mettre à jour.
10. Choisissez Next pour chacune des pages suivantes jusqu'à ce que la page Révision et mise à disposition apparaisse. Sur la page Révision et mise à disposition, sélectionnez à la fois Je reconnais que cela AWS CloudFormation peut créer des ressources IAM avec des noms personnalisés et je comprends que les modifications apportées au modèle sélectionné peuvent entraîner la mise AWS CloudFormation à jour ou la suppression de AWS ressources existantes.
11. Sélectionnez Approvisionner une pile.

Si vous devez annuler vos mises à jour, procédez comme suit.

1. Choisissez Actions, puis Provision stack.
2. Choisissez l'une des versions existantes, puis choisissez la version précédente du modèle.
3. Choisissez Sélectionner une pile existante, puis choisissez la CloudFormation pile à mettre à jour.

Les recettes de configuration de livres de recettes personnalisées sont-elles migrées ?

L'exécution de livres de recettes personnalisés n'est pas prise en charge lors d'un événement d'installation. Le script migre les recettes de configuration personnalisées des livres de recettes et crée un runbook Systems Manager Automation pour vous. Cependant, vous devez exécuter les recettes manuellement.

Procédez comme suit pour exécuter vos recettes de configuration.

1. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.
2. Dans le volet de navigation, choisissez Application Manager.
3. Dans la section Applications, sélectionnez Applications personnalisées.
4. Choisissez votre application. Le nom de l'application commence par `app-`*`stack-name`*.

5. Choisissez Resources, puis choisissez le runbook de configuration.
6. Choisissez Execute Automation.
7. Choisissez l'instance IDs pour laquelle vous souhaitez exécuter les recettes de configuration, puis choisissez Execute.

Puis-je exécuter des recettes de déploiement et de dédéploiement sur mes instances nouvellement créées ?

Le script peut créer trois runbooks d'automatisation possibles en fonction de la configuration de votre couche.

- Configuration
- Configuration
- Terminer

Le script peut également créer les paramètres Systems Manager suivants qui contiennent des valeurs d'entrée pour le `AWS-ApplyChefRecipes Run Command` document.

- Configuration
- Déploiement
- Configuration
- Annulation du déploiement
- Terminer

Lorsqu'un événement de scale-out se produit, le runbook de configuration Automation s'exécute automatiquement. Cela inclut la configuration et le déploiement de recettes de livres de recettes personnalisées à partir de votre OpsWorks couche d'origine. Lorsqu'un événement de scale-in se produit, le runbook Terminate Automation s'exécute automatiquement. Le runbook Terminate Automation contient les recettes d'arrêt de votre OpsWorks couche d'origine.

Si vous souhaitez exécuter des recettes d'annulation ou de configuration manuellement, procédez comme suit.

1. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.

2. Dans le volet de navigation, choisissez Application Manager.
3. Dans la section Applications, sélectionnez Applications personnalisées.
4. Choisissez votre application. Le nom de l'application commence par `app-`*`stack-name`*`-`*`first-six-characters-stack-id`*. Le gestionnaire d'applications ouvre l'onglet Vue d'ensemble.
5. Choisissez Resources, puis choisissez le runbook de configuration de l'automatisation.
6. Choisissez Execute Automation.
7. Pour le paramètre d'entrée `applyChefRecipesPropertiesParameter` Automation Runbook, référencez le paramètre Systems Manager correct. Le nom du paramètre Systems Manager suit le format `/ApplyChefRecipes-Preset/`*`OpsWorks-stack-name`*`-`*`OpsWorks-layer-name`*`-`*`first-six-characters-stack-id`*`/`*`event`*, dans lequel la valeur *`event`* est `ConfigureDeploy`, ou `Undeploy` en fonction des recettes que vous souhaitez exécuter.
8. Choisissez l'instance IDs dans laquelle vous souhaitez exécuter les recettes, puis choisissez Execute.

Puis-je modifier les sous-réseaux couverts par mon groupe Auto Scaling ?

Par défaut, le groupe Auto Scaling couvre tous les sous-réseaux de votre stack OpsWorks VPC. Pour mettre à jour les sous-réseaux à couvrir, procédez comme suit.

1. Choisissez `Link to the template` affiché dans la sortie du script. Si vous avez fermé votre terminal, procédez comme suit pour accéder au modèle.
   a. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.
   b. Dans le volet de navigation, choisissez Application Manager.
   c. Choisissez des CloudFormation piles, puis choisissez Bibliothèque de modèles.
   d. Choisissez Owned by me et localisez votre modèle.
2. Dans Actions, sélectionnez Provision stack.
3. Confirmez que vous souhaitez utiliser le modèle par défaut. Choisissez Sélectionner une pile existante, puis choisissez la CloudFormation pile à mettre à jour.

> **Note**
>
> Si vous avez exécuté le script avec le `--provision-application` paramètre défini sur`FALSE`, vous devez créer une nouvelle CloudFormation pile.

4. Pour le `SubnetIDs` paramètre, fournissez une liste séparée par des virgules des sous-réseaux IDs que vous souhaitez étendre à votre groupe Auto Scaling.
5. Choisissez Next jusqu'à ce que la page Révision et approvisionnement apparaisse.
6. Sur la page Révision et mise à disposition, choisissez Je reconnais que des ressources IAM AWS CloudFormation peuvent être créées avec des noms personnalisés. Je comprends que les modifications apportées au modèle sélectionné peuvent entraîner la mise AWS CloudFormation à jour ou la suppression de AWS ressources existantes.
7. Sélectionnez Approvisionner une pile.

Résolution des problèmes

Cette section présente certains problèmes courants et propose des solutions à ces problèmes.

Rubriques

- Le principal fourni n'est pas valide
- Impossible de supprimer la CloudFormation pile lorsque les instances protégées par le groupe Auto Scaling sont activées
- Erreur d'accès refusé lors de la fourniture d'un compartiment et d'un préfixe S3 existants

Le principal fourni n'est pas valide

Problème : vous recevez un message d'erreur indiquant que le principal que vous avez indiqué n'est pas valide.

Cause : Cela se produit parce que le groupe Auto Scaling n'a pas de rôle de service.

Solution : créez un groupe Auto Scaling dans la région où l'erreur s'est produite. La création d'un groupe Auto Scaling crée le rôle lié au service nécessaire pour votre politique de résiliation personnalisée.

Impossible de supprimer la CloudFormation pile lorsque les instances protégées par le groupe Auto Scaling sont activées

Problème : le `--enable-instance-protection` paramètre est défini sur `TRUE` et certaines EC2 instances de votre groupe Auto Scaling sont protégées par la clé `protected_instance` tag, ce qui empêche la suppression complète de votre AWS CloudFormation stack.

Cause : Les EC2 instances disposent d'une clé de `protected_instance` balise qui les protège des événements de résiliation.

Solution : supprimez la clé de `protected_instance` balise des EC2 instances. Cela permet au groupe Auto Scaling de réduire sa taille. Une fois que le groupe Auto Scaling a été réduit, vous pouvez supprimer la AWS CloudFormation pile.

Erreur d'accès refusé lors de la fourniture d'un compartiment et d'un préfixe S3 existants

Problème : vous recevez un `AccessDenied` message d'erreur lorsque vous fournissez un compartiment et un préfixe S3 existants.

Cause : La politique du compartiment S3 ne fournit pas les autorisations nécessaires pour transmettre les journaux de l'équilibreur de charge au compartiment.

Solution : mettez à jour la politique du compartiment S3 pour permettre au script de fournir les journaux d'accès de l'équilibreur de charge au compartiment. Pour plus d'informations sur la mise à jour de la politique relative aux compartiments, consultez la section Activer les journaux d'accès pour votre Application Load Balancer dans le guide de l'utilisateur d'Elastic Load Balancing : Application Load Balancers.

Utilisation de l' AWS OpsWorks Stacks outil Detach in Place

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible.

Cette section décrit comment utiliser l'outil AWS OpsWorks Stacks Detach in Place pour détacher vos OpsWorks instances du service OpsWorks Stacks.

Les instances que vous détachez resteront dans le vôtre Compte AWS, mais vous ne pourrez plus les gérer à l'aide OpsWorks de. Vous utiliserez plutôt Amazon EC2 ou toute autre approche EC2 compatible pour configurer et gérer les instances. AWS Systems Manager

À un niveau élevé, le processus de détachement comprend les étapes suivantes :

1. L'outil effectue des contrôles de validation pour s'assurer que les ressources sont prêtes à être détachées.
2. L'outil exporte le code JSON personnalisé depuis votre OpsWorks pile et le stocke sous forme d'objet dans Amazon S3.
3. L'outil crée des documents Systems Manager Automation représentant chaque événement du cycle de vie de OpsWorks Stacks.
4. L'outil crée un AWS Service Catalog AppRegistry catalogue pour toutes les instances qui sont détachées et détache les équilibreurs de charge Elastic Load Balancing (ELB) des couches. OpsWorks
5. Enfin, l'outil détache et désenregistre d'autres ressources, notamment les instances Amazon Relational Database Service (Amazon RDS).

Comment fonctionne le processus

L'outil Detach In Place fournit les 3 commandes suivantes et une expérience semblable à celle d'un assistant qui vous guide à travers une série d'étapes pour vérifier et configurer vos instances avant de procéder au détachement de votre couche.

Command	Description
`handle-prerequisites`	Cette commande analyse si toutes les instances d'une couche sont éligibles au détachement et résout les conditions préalable s. Les instances doivent être en bon état OpsWorks, elles ne peuvent pas être équipées de scalers automatiques basés sur le temps ou la charge, et la dernière version de l' OpsWorks agent doit être installée.

Command	Description
	En outre, la commande vérifie si toutes les instances disposent des autorisations requises pour prendre en charge l'agent SSM et si la dernière version de l'agent SSM est installée. La commande installera l'agent SSM s'il n'est pas présent et mettra à jour l'agent SSM s'il n'utilise pas la dernière version. La commande ajoutera également les autorisations nécessair es.

Command	Description
`detach`	Cette commande détache toutes les OpsWorks instances de la couche spécifiée. Tout d'abord, la commande exécutera une vérification des conditions préalable s pour s'assurer que la couche est éligible au détachement. Si vous ne souhaitez pas résoudre les conditions préalables, vous avez la possibilité de forcer le détachement. Ensuite, la commande indiquera que toutes les balises ajoutées à vos instances par le biais du OpsWorks balisage APIs ou de la propagation de balises à partir de vos couches et piles seront conservées. Vous pouvez supprimer n'importe laquelle de ces balises à l'aide des EC2 APIs outils appropriés une fois le détachement terminé. Ensuite, la commande vérifiera si vous souhaitez exporter la configuration associée à Chef vers les paramètres SSM. Si un Classic Load Balancer est attaché à la couche, la commande vous demandera s'il peut détacher l'équilibreur de charge afin d'éviter tout temps d'arrêt.

Command	Description
`cleanup`	Cette commande supprime toutes les entités OpsWorks de votre compte. Cela mettra fin aux instances et supprimera toutes les piles. Cela doit être utilisé pour les ressources qui ne sont plus nécessaires comme dernière étape pour nettoyer le compte. **Note** Nous vous recommandons d'exécuter la nouvelle installation pendant quelques jours avant d'exécuter la `cleanup` commande. Cela garantit que toutes les configurations nécessaires issues de la pile sont facilement disponibles en cas de besoin.

Limites

L'objectif principal de l'outil Detach In Place est de détacher les instances OpsWorks Stacks en toute sécurité. Cette section résume les limites de l'outil.

- Agent SSM Windows : si l'agent SSM n'est pas installé sur l'instance, vous devez l'installer manuellement. Il en va de même si l'agent n'est pas mis à jour vers la dernière version.
- Instances Time/Load Auto Scaling : l'outil de détachement ne prend pas en charge les instances pour lesquelles Auto Scaling est activé. Vous devez désactiver Auto Scaling sur les instances que vous souhaitez détacher.
- Autorisations : l'outil de détachement ne crée ni ne génère les entités IAM spécifiées sur la page Autorisations de la OpsWorks console.
- Applications : l'outil de détachement ne crée ni ne génère d'applications en dehors de OpsWorks.

Premiers pas

Étape 1 : vérifier que les conditions préalables sont remplies

Les 3 commandes de l'outil Detach In Place sont des scripts Python, que vous pouvez exécuter localement, sur EC2 instance ou en utilisant AWS CloudShell.

AWS CloudShell est un shell basé sur un navigateur qui vous permet d'accéder en ligne de commande aux AWS ressources du fichier sélectionné. Région AWS AWS CloudShell est préinstallé avec des outils populaires (tels que AWS CLI Python). Lors de l'utilisation AWS CloudShell, vous utilisez les mêmes informations d'identification que celles que vous utilisez pour vous connecter à la console.

Cette procédure pas à pas suppose que vous utilisez AWS CloudShell.

Étape 2 : Téléchargez le script

1. Téléchargez le fichier zip qui contient le script de migration et tous les fichiers pertinents en exécutant la commande suivante :

   ```
   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. Décompressez le fichier en exécutant la commande suivante.

   ```
   unzip detach_in_place_script.zip
   ```

   Une fois le fichier décompressé, les fichiers suivants sont disponibles :

   - Lisez-moi .md
   - LICENCE
   - NOTICE
   - requirements.txt
   - TODO.py

3. Si nécessaire, installez `pipenv` en exécutant la commande suivante.

   ```
   pip install pipenv
   ```

Étape 3 : Exécuter le script

Configurez d'abord votre environnement de manière à pouvoir exécuter le script en exécutant les commandes suivantes.

```
pipenv install -r requirements.txt
pipenv shell
```

Passez ensuite en revue les paramètres du script.

Command	Paramètre	Description	Type	Obligatoire	Par défaut
`handle-prerequisites`	`--layer-id`	ID de la couche que vous souhaitez détacher.	Chaîne	Oui	-
	`--region`	La région de la OpsWorks pile. Si votre région de OpsWorks pile et votre région de point de terminaison d'API sont différentes, utilisez la région de pile. Il s'agit de la même région que les autres ressources de votre OpsWorks pile (par exemple, les EC2 instances et les sous-réseaux).	Chaîne	Non	us-east-1
`detach`	`--layer-id`	ID de la couche que vous souhaitez détacher.	Chaîne	Oui	-
	`--batch-size`	Nombre d'instances à détacher d'une couche (par exemple, 5).	Chaîne	Non	-
	`--region`	La région de la OpsWorks pile. Si votre région	Chaîne	Non	us-east-1

Command	Paramètre	Description	Type	Obligatoire	Par défaut
		de OpsWorks pile et votre région de point de terminaison d'API sont différentes, utilisez la région de pile. Il s'agit de la même région que les autres ressources de votre OpsWorks pile (par exemple, les EC2 instances et les sous-réseaux).			
`cleanup`	`--stack-id`	ID de la pile que vous souhaitez supprimer.	Chaîne	Non	S'excluant mutuellement, vous devez spécifier un ID de couche ou un ID de pile
	`--layer-id`	ID de la couche que vous souhaitez supprimer	Chaîne	Non	

Command	Paramètre	Description	Type	Obligatoire	Par défaut
	`--region`	La région de la OpsWorks pile. Si votre région de OpsWorks pile et votre région de point de terminaison d'API sont différentes, utilisez la région de pile. Il s'agit de la même région que les autres ressources de votre OpsWorks pile (par exemple, les EC2 instances et les sous-réseaux).	Chaîne	Non	us-east-1

Vous pouvez voir les options disponibles pour les `cleanup` commandes`detach`, `handle-prerequisites` et en exécutant les commandes avec l'`--help`option comme suit :

```
python3 layer_detacher.py detach --help
python3 layer_detacher.py handle-prerequisites --help
python3 layer_detacher.py cleanup --help
```

Vous êtes maintenant prêt à commencer. Les exemples suivants montrent comment exécuter les commandes pour différents cas d'utilisation.

Exemples :

- Exemple 1 : vérifier si une couche remplit toutes les conditions requises et est éligible au détachement
- Exemple 2 : détacher toutes les instances d'une couche
- Exemple 3 : détacher toutes les instances d'une couche par lots
- Exemple 4 : nettoyer toutes les ressources d'une couche et supprimer la couche
- Exemple 5 : nettoyer toutes les ressources d'une pile et supprimer la pile

Exemple 1 : vérifier si une couche remplit toutes les conditions requises et est éligible au détachement

La commande suivante lit les informations relatives à une OpsWorks couche (et aux instances qu'elle inclut) et vérifie si les conditions préalables suivantes sont remplies :

- Toutes les instances sont en ligne.
- Il n'existe aucune instance Load/Time Auto Scaling.
- Toutes les instances disposent de la dernière version de OpsWorks l'agent.
- La dernière version de l'agent SSM est installée et configurée sur toutes les instances.
- Toutes les instances possèdent une paire de clés SSH.
- Chaque instance appartient exactement à une couche.

```
python3 layer_detacher.py handle-prerequisites \
--layer-id opsworks-layer-id \
--region opsworks-stack-region
```

Exemple 2 : détacher toutes les instances d'une couche

La commande suivante va itérer sur toutes les instances de la couche, vérifier si les instances répondent aux prérequis et essayer de détacher en parallèle toutes les instances qui répondent aux prérequis. Si un ou plusieurs prérequis ne sont pas remplis, la commande fournira une option de détachement forcé pour les instances non conformes restantes.

Avant de détacher une instance, la commande doit :

1. Enregistrez le code JSON personnalisé et téléchargez-le dans S3.
2. Créez des documents d'automatisation SSM pour chaque événement OpsWorks du cycle de vie de la couche et téléchargez les journaux d'exécution des documents d'automatisation sur S3.
3. Créez une AppRegistry application pour toutes les instances qui seront détachées. L'application est associée à un groupe de ressources qui contient toutes les instances et ressources détachées. Les ressources incluent des documents SSM Automation et des paramètres SSM contenant des informations sur les événements du cycle de vie et des recettes Chef personnalisées.
4. Détache le Classic Load Balancer de la couche, s'il en existe un.

Cette commande modifiera uniquement OpsWorks les ressources. Le statut des EC2 instances restera le même.

```
python3 layer_detacher.py detach \
--layer-id opsworks-layer-id \
--region opsworks-stack-region
```

Exemple 3 : détacher toutes les instances d'une couche par lots

La commande suivante fait la même chose que dans l'exemple précédent. La seule différence est qu'il détache les instances par lots.

Cette commande modifiera uniquement OpsWorks les ressources. Le statut des EC2 instances restera le même.

```
python3 layer_detacher.py detach \
--layer-id opsworks-layer-id \
--region opsworks-stack-region \
--batch-size 5
```

Exemple 4 : nettoyer toutes les ressources d'une couche et supprimer la couche

La commande suivante va itérer toutes les ressources d'une couche et les supprimer. Plus en détail, il arrêtera et supprimera toutes les instances présentes dans OpsWorks et EC2, détachera l'équilibreur de charge et annulera l'enregistrement des instances Amazon RDS, Elastic et des volumes. IPs Après avoir nettoyé les ressources, la couche sera supprimée.

Cette commande supprimera les OpsWorks ressources et les EC2 instances. Si vous souhaitez que vos EC2 instances restent intactes, utilisez la `detach` commande avant de l'`cleanup`utiliser. De cette façon, la `cleanup` commande supprimera toutes les ressources restantes.

```
python3 layer_detacher.py cleanup \
--layer-id opsworks-layer-id \
--region opsworks-stack-region
```

Exemple 5 : nettoyer toutes les ressources d'une pile et supprimer la pile

La commande suivante va itérer sur toutes les couches, puis sur les ressources de chaque couche. Pour chaque couche, la commande arrête et supprime toutes les instances présentes dans OpsWorks et EC2, détache les équilibreurs de charge et annule l'enregistrement des instances Amazon RDS, Elastic et Volumes. IPs Ensuite, la commande supprimera la couche. Le même

processus sera effectué dans chaque couche appartenant à cette pile. Enfin, une fois toutes les couches supprimées, la pile sera supprimée.

Cette commande supprimera les OpsWorks ressources et les EC2 instances. Si vous souhaitez que vos EC2 instances restent intactes, utilisez la `detach` commande avant de l'`cleanup`utiliser. De cette façon, la `cleanup` commande supprimera toutes les ressources restantes.

```
python3 layer_detacher.py cleanup \
--stack-id opsworks-stack-id \
--region opsworks-stack-region
```

Étape 4 : Continuez à exploiter vos ressources après vous être détaché de OpsWorks

Après avoir exécuté la `detach` commande, l'outil crée une nouvelle AWS Service Catalog AppRegistry application correspondant à la couche détachée. Le nom de l'application suit le format*`layer-name`*`---`*`layer-id`*. Il ajoute également la `OpsWorksLayerId` balise pour identifier de manière unique l'application correspondant à la couche détachée.

Pour ajouter de nouvelles AWS ressources à cette application (par exemple, de nouvelles EC2 instances), vous pouvez effectuer l'une des opérations suivantes :

1. Marquez la ressource avec le tag d'application unique de l' AppRegistryapplication :

   Clé du tag : `awsApplication`

   Valeur : `arn:aws:resource-groups:`*`region`*`:`*`account-id`*`:group/`*`application-name`*`/`*`application-id`*`>`
2. Exécutez le associate-resourcecommande.

En outre, pour chaque AppRegistry application, un groupe de ressources est créé. Le groupe de ressources contient les balises suivantes.

Clé de balise	Valeur
`EnableAWSServiceCatalogAppRegistry`	`TRUE`
`aws:servicecatalog:applicationName`	*`application-name`*

Clé de balise	Valeur
`aws:servicecatalog:applicationId`	*application-id*
`aws:servicecatalog:applicationArn`	`arn:aws:servicecatalog:` *region*`:`*account-id*` :/applications/` *application-id*

Exécution de tâches après le détachement

Le tableau suivant fournit des informations sur la manière d'effectuer des tâches après le détachement :

Tâche	Description
Exécution d'événements liés au cycle	Après avoir exécuté la `detach` commande et si vous avez sélectionné l'option, le script crée 5 documents d'automatisation correspondant aux 5 événements OpsWorks du cycle de vie. Le nom de chaque document d'automatisation suit ce format :*layer-id*`_`*lifecycle-event*`_automation_document` . Pour simuler OpsWorks le comportement dans Systems Manager, vous devez déclencher manuellement des exécutions d'automatisation lors du provisionnement, de la résiliation d' EC2 instances ou du déploiement/de la suppression de recettes.
Mise à jour du JSON personnalisé	Les fichiers JSON personnalisés pour la pile et la couche sont stockés dans un compartiment S3 spécifié lors du détachement, ou bien dans un nouveau compartiment S3 créé. Les noms de fichiers enregistrés pour les fichiers JSON sont les suivants :

Tâche	Description
	• `layercustomjson.json` • `stackcustomjson.json`
Modification de votre liste de courses pour les événements du cycle de vie	La liste des exécutions pour chaque événement du cycle de vie est définie dans le document d'automatisation correspondant. Pour modifier la liste des exécutions, recherche z les documents d'automatisation dans l' AppRegistry application et modifiez le `RunList` paramètre. Le processus de mise à jour des recettes et des livres de recettes est inchangé car`AWS-ApplyChefRecipes` , celui déclenché par les documents d'automatisation, prend en charge la même source que OpsWorks.
Gestion de la guérison automatique et de la mise à l'échelle automatique	Lorsque vous détachez une instance, l' OpsWorks agent est désinstallé. Sans l'agent, vous OpsWorks ne pouvez pas réparer ou remplacer automatiquement les instances défectueuses, pas plus qu'il ne peut faire évoluer automatiquement votre flotte. Pour continuer à effectuer le dimensionnement automatique et à remplacer les instances défaillantes, créez un groupe Amazon EC2 Auto Scaling. Le groupe lancera de nouvelles instances pour maintenir la capacité souhaitée lorsqu'Amazon EC2 détectera des instances défectueuses devant être remplacées.

Tâche	Description
Gestion du Load Balancer	Si votre couche utilise un Classic Load Balancer, la `detach` commande le détachera avant de désenregistrer les instances. Cela permet de garantir que toutes les associati ons d'instances ELB restent préservées sur Amazon EC2 tout au long du détachement, ce qui permet d'éviter toute interruption de service. Une fois le processus terminé, vous pourrez gérer votre ELB sur EC2.
Connexion aux instances	Lorsque vous exécutez la `detach` commande `handle-prerequisites` or, deux vérificat ions sont effectuées : • Version de l'agent SSM et autorisations • Clés SSH Les commandes vous offrent également la possibilité de mettre à jour l'agent SSM et d'ajouter les autorisations requises afin que vous puissiez vous connecter aux instances à l'aide du gestionnaire de session. Si des clés SSH existent, vous avez également la possibili té d'accéder à l'instance en SSH.

Utilisation de l'onglet Instances du gestionnaire d'applications Systems Manager

Après le détachement, vous pourrez consulter et gérer vos instances dans l'onglet Instances d'Application Manager.

L'onglet Instances fournit des informations agrégées sur les EC2 instances d'une application, telles que leur statut, leur état de santé et le statut de la dernière commande. À l'aide de cet onglet, vous pouvez consulter des informations détaillées sur les instances individuelles, telles que l'historique des commandes, les états d'alarme, l'état de santé des agents Systems Manager, etc. L'onglet Instances propose également diverses actions, telles que la possibilité d'appliquer des recettes Chef,

de démarrer ou d'arrêter une instance, ou d'ajouter ou de supprimer une instance d'un groupe Auto Scaling.

Débuter avec AWS OpsWorks Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks fournit un ensemble complet de composants personnalisables que vous pouvez combiner pour créer une pile répondant à vos objectifs spécifiques. Le défi pour les nouveaux utilisateurs est de comprendre comment assembler ces éléments dans une pile de travail et gérer celle-ci efficacement. Voici comment vous pouvez commencer.

Si vous souhaitez...	Effectuez cette procédure :
Créer un exemple de pile aussi rapidement que possible	Mise en route : exemple
Expérimenter avec une pile Linux	Mise en route : Linux
Expérimenter avec une pile Windows	Mise en route : Windows
Découvrez comment créer vos propres livres de recettes Chef	Mise en route : livres de recettes

Si vous disposez de ressources informatiques existantes (EC2 instances Amazon ou même instances sur site exécutées sur votre propre matériel), vous pouvez les intégrer dans une pile, ainsi que les instances que vous avez créées avec Stacks. AWS OpsWorks Vous pouvez ensuite utiliser AWS OpsWorks Stacks pour gérer toutes les instances associées en tant que groupe, quelle que soit la manière dont elles ont été créées.

Prise en charge de la région

Vous pouvez accéder à AWS OpsWorks Stacks dans le monde entier ; vous pouvez également créer et gérer des instances dans le monde entier. Les utilisateurs peuvent configurer les instances AWS OpsWorks Stacks pour qu'elles soient lancées dans n'importe quelle AWS région, à l'exception AWS GovCloud de l'ouest des États-Unis et de la région de Chine (Pékin). Pour fonctionner avec AWS OpsWorks Stacks, les instances doivent pouvoir se connecter à l'un des points de terminaison de l'API du service d'instance AWS OpsWorks Stacks suivants.

Les ressources peuvent être gérées uniquement dans la région dans laquelle elles sont créées. Les ressources qui sont créées dans un point de terminaison régional ne sont pas disponibles et ne peuvent pas être clonées dans un autre point de terminaison régional. Vous pouvez lancer des instances dans l'une des régions suivantes.

- Région US East (Ohio)
- Région USA Est (Virginie du Nord)
- Région USA Ouest (Oregon)
- Région US West (N. California)
- Région du Canada (Centre) (API uniquement, non disponible pour les piles créées dans le AWS Management Console.)
- Région Asie-Pacifique (Mumbai)
- Région Asia Pacific (Singapore)
- Région Asie-Pacifique (Sydney)
- Région Asia Pacific (Tokyo)
- Région Asia Pacific (Seoul)
- Région Europe (Frankfurt)
- Région Europe (Irlande)
- Région Europe (Londres)
- Région Europe (Paris)
- Région Amérique du Sud (São Paulo)

Mise en route avec un exemple de pile

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette présentation explique comment utiliser AWS OpsWorks Stacks pour créer rapidement un exemple d'environnement d'application Node.js en quelques clics de souris et sans écrire de code. Lorsque vous avez terminé, vous disposez d'une instance Amazon Elastic Compute Cloud (Amazon EC2) exécutant Chef 12, d'un serveur HTTP Node.js et d'une application Web que vous pouvez utiliser pour interagir avec Twitter et laisser des commentaires sur une page Web.

> **Note**
>
> Comme cette procédure pas à pas crée automatiquement une instance de type c3.large, vous ne pouvez pas utiliser cette procédure pas à pas, ni l'outil de création de Sample Stack dans AWS OpsWorks Stacks, dans le niveau gratuit.AWS Bien que l'utilisation de l'outil de création Sample Stack dans un VPC crée une instance t2.medium, elle n'est VPCs actuellement pas disponible dans le niveau gratuit.AWS

Étape 1 : Exécuter les opérations prérequises

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous devez effectuer les étapes de configuration suivantes avant de commencer la procédure. Ces étapes de configuration incluent la création d'un AWS compte, la création d'un utilisateur administratif et l'attribution d'autorisations d'accès à AWS OpsWorks Stacks.

Rubriques

- Inscrivez-vous pour un Compte AWS
- Création d'un utilisateur doté d'un accès administratif
- Attribuer des autorisations d'accès au service

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

   Une partie de la procédure d'inscription consiste à recevoir un appel téléphonique ou un message texte et à saisir un code de vérification sur le clavier du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à https://aws.amazon.com/et en choisissant Mon compte.

Création d'un utilisateur doté d'un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez Utilisateur racine d'un compte AWS AWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez Connexion en tant qu'utilisateur racine dans le Guide de l'utilisateur Connexion à AWS .
2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, voir Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d' AWS IAM Identity Center dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu'utilisateur doté d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

  Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribution d'un accès à d'autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d'autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

Pour obtenir des instructions, consultez Création d'un ensemble d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

   Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l'utilisateur AWS IAM Identity Center .

Attribuer des autorisations d'accès au service

Activez l'accès au service AWS OpsWorks Stacks (et aux services connexes sur lesquels AWS OpsWorks Stacks s'appuie) en ajoutant les `AmazonS3FullAccess` autorisations `AWSOpsWorks_FullAccess` et à votre rôle ou à votre utilisateur.

Pour plus d'informations sur l'ajout d'autorisations, consultez la section Ajout d'autorisations d'identité IAM (console).

Vous avez maintenant terminé toutes les étapes de configuration et pouvez démarrer cette procédure pas à pas.

Étape 2 : Créer une pile

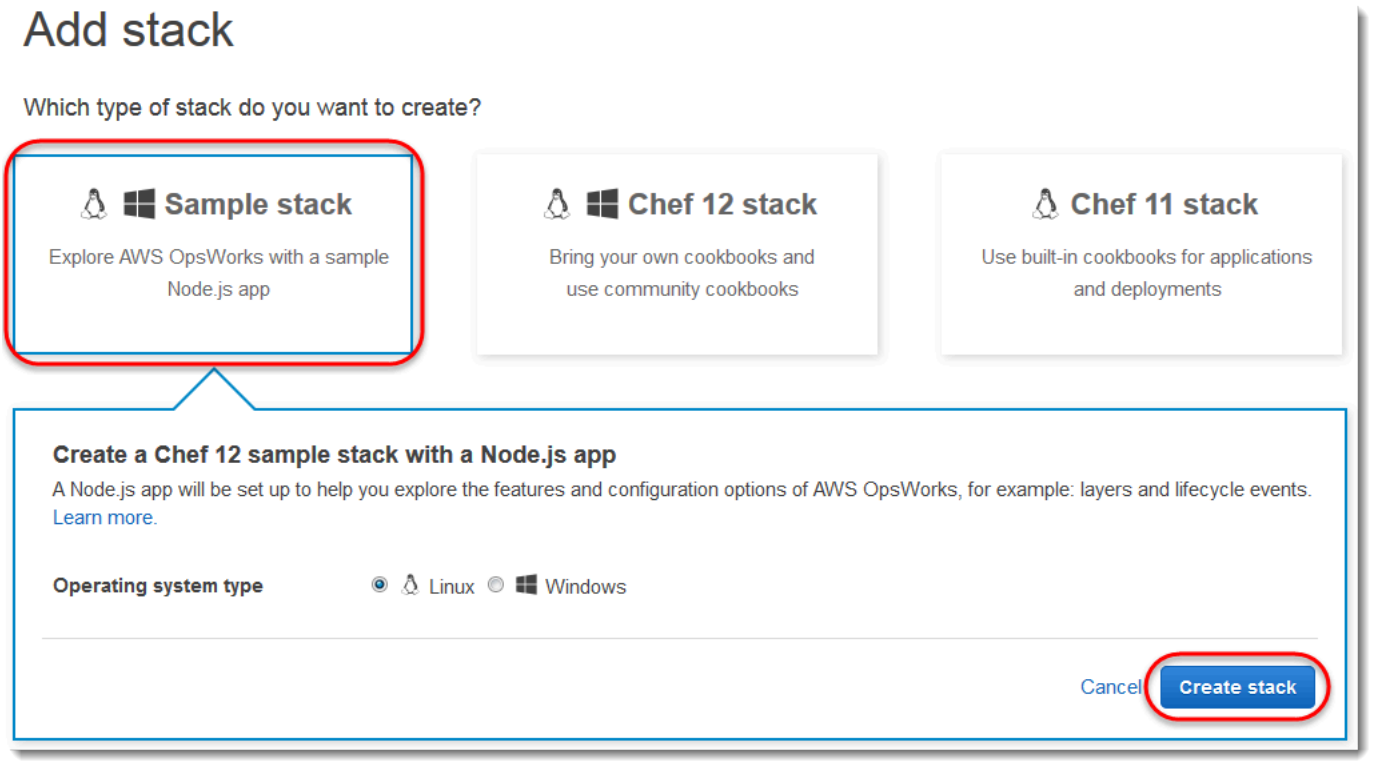
> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Dans cette étape, vous allez utiliser la console AWS OpsWorks Stacks pour créer une pile. Une pile est un ensemble d'instances (telles que les EC2 instances Amazon) et de AWS ressources associées qui ont un objectif commun et que vous souhaitez gérer ensemble. (Pour plus d'informations, consultez Piles.) Il n'y aura qu'une seule instance pour cette procédure.

Avant de commencer cette étape, remplissez les conditions préalables.

Pour créer la pile

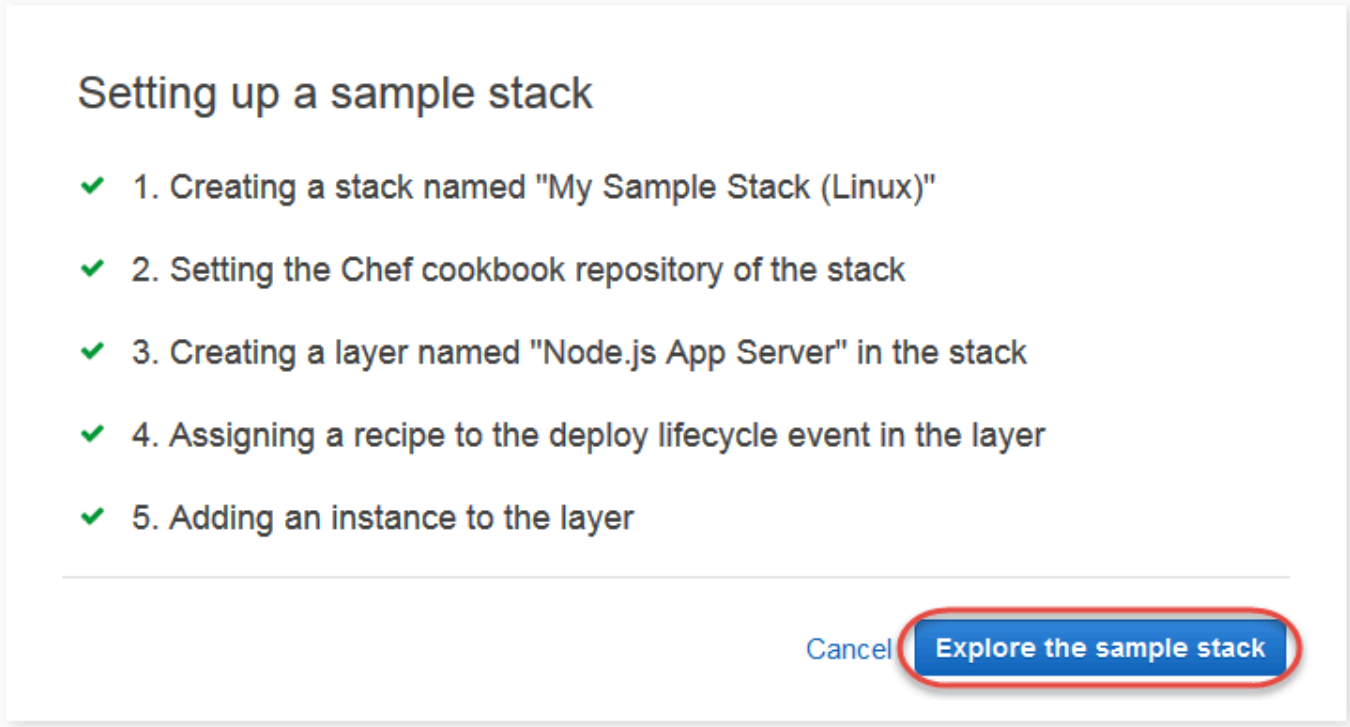
1. Connectez-vous à la AWS OpsWorks console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/opsworks/.
2. Exécutez l'une des actions suivantes, si elles s'appliquent :
   - Si la page Bienvenue dans AWS OpsWorks Stacks s'affiche, choisissez Ajouter votre première pile ou Ajouter votre première pile AWS OpsWorks Stacks (les deux choix ont le même effet). La page Add stack (Ajouter une pile) s'affiche.
   - Si la page OpsWorks Tableau de bord s'affiche, choisissez Ajouter une pile. La page Add stack (Ajouter une pile) s'affiche.
3. Avec la page Add stack (Ajouter une pile) affichée, choisissez Sample stack (Exemple de pile) si ce n'est déjà fait.
4. Avec Linux déjà choisi pour Operating system type (Type de système d'exploitation), choisissez Create stack (Créer une pile) :



5. AWS OpsWorks Stacks crée une pile nommée My Sample Stack (Linux). AWS OpsWorks Stacks ajoute également tous les composants nécessaires pour déployer l'application dans la pile :

- Une couche, qui est un plan pour un ensemble d'instances. Elle spécifie les éléments tels que les paramètres, les ressources, les packages installés et les groupes de sécurité de l'instance. (Pour plus d'informations, consultez Couches.) La couche s'appelle Node.js App Server.
- Une instance, qui dans ce cas est une EC2 instance Amazon Linux 2. (Pour plus d'informations sur les instances , consultez la section instances.) Le nom d'hôte de l'instance est nodejs-server1.
- Une application, c'est-à-dire le code à exécuter sur l'instance. (Pour plus d'informations sur les applications, consultez Applications.) L'application s'appelle Node.js Sample App (Exemple d'application Node.js).

6. Une fois que AWS OpsWorks Stacks a créé la pile, choisissez Explore the sample stack (Linux) pour afficher la page My Sample Stack (Linux) (si vous effectuez cette procédure pas à pas plusieurs fois, My Sample Stack (Linux) peut être suivi d'un numéro séquentiel, tel que 2 ou 3) :



Au cours de l'étape suivante, vous démarrerez l'instance et vous déploierez l'application sur l'instance.

Étape 3 : Redémarrer l'instance et déployer l'application

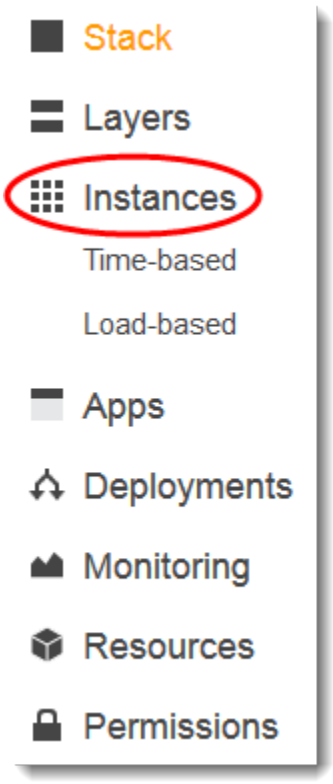
> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Maintenant que vous avez une instance et une application, démarrez l'instance et déployez l'application sur celle-ci.

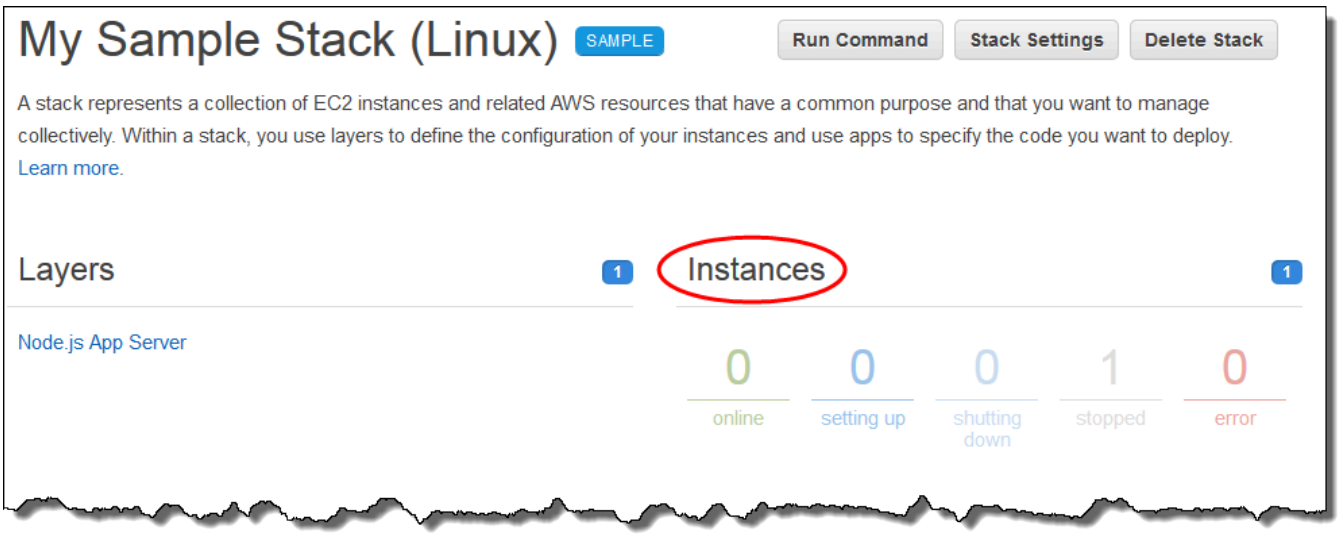
Pour démarrer l'instance et déployer l'application

1. Effectuez l'une des actions suivantes :

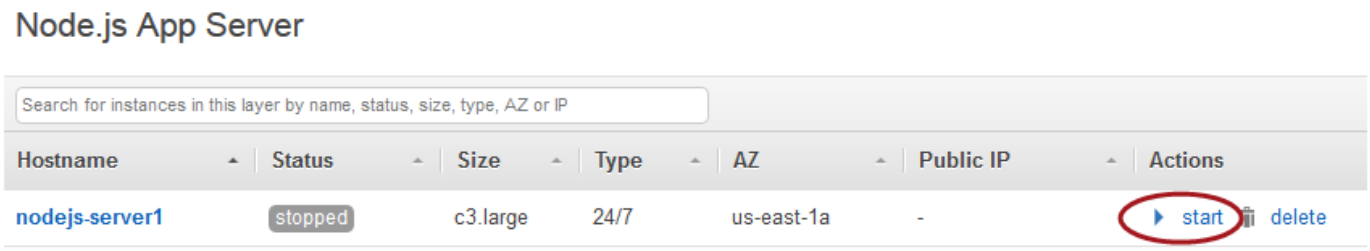
   - Dans le panneau de navigation du service, choisissez Instances :



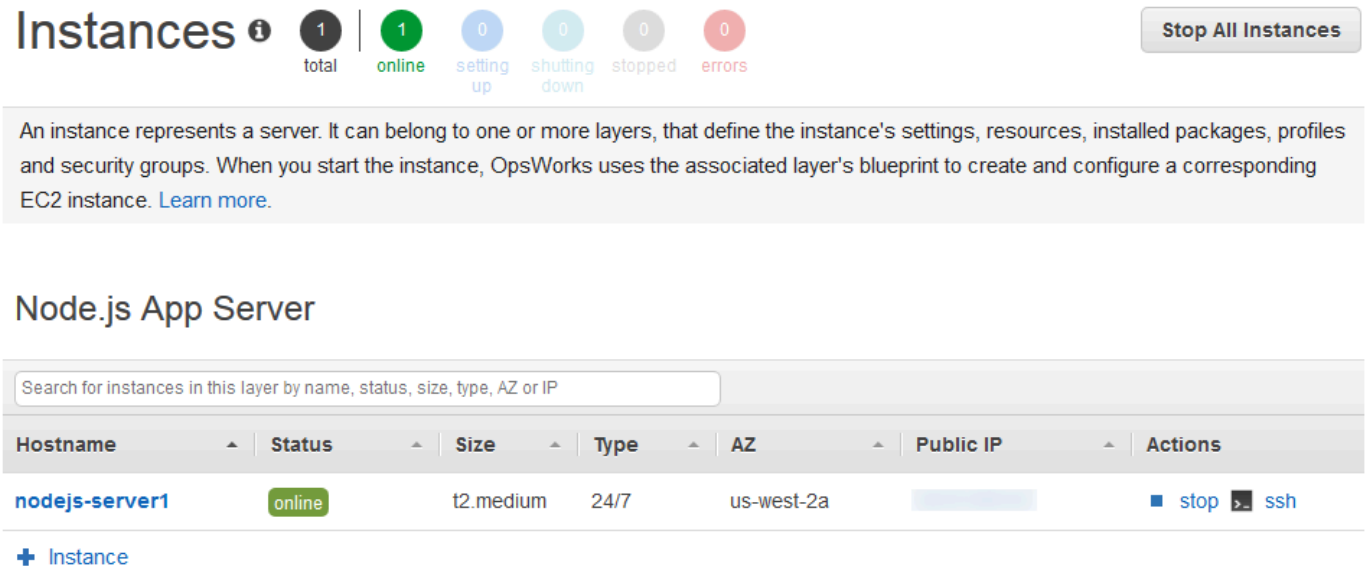
- Sur la page My Sample Stack (Linux) [Mon exemple de pile (Linux)], choisissez Instances :



2. Sur la page Instances, pour Node.js App Server, pour nodejs-server1, choisissez start (démarrer) :



3. Poursuivez uniquement lorsque l'indicateur circulaire online (en ligne) est vert vif. (Si vous voyez un message d'erreur, consultez Guide de débogage et dépannage.)
4. Au fur et à mesure de la configuration de l'instance, AWS OpsWorks Stacks déploie l'application sur l'instance.
5. Vos résultats doivent ressembler à la capture d'écran suivante avant que vous ne continuiez (si vous recevez un message d'erreur, vous pouvez consulter le Guide de débogage et dépannage) :



Vous avez maintenant une instance avec une application qui a été déployée dessus.

Au cours de l'étape suivante, vous allez tester l'application sur l'instance.

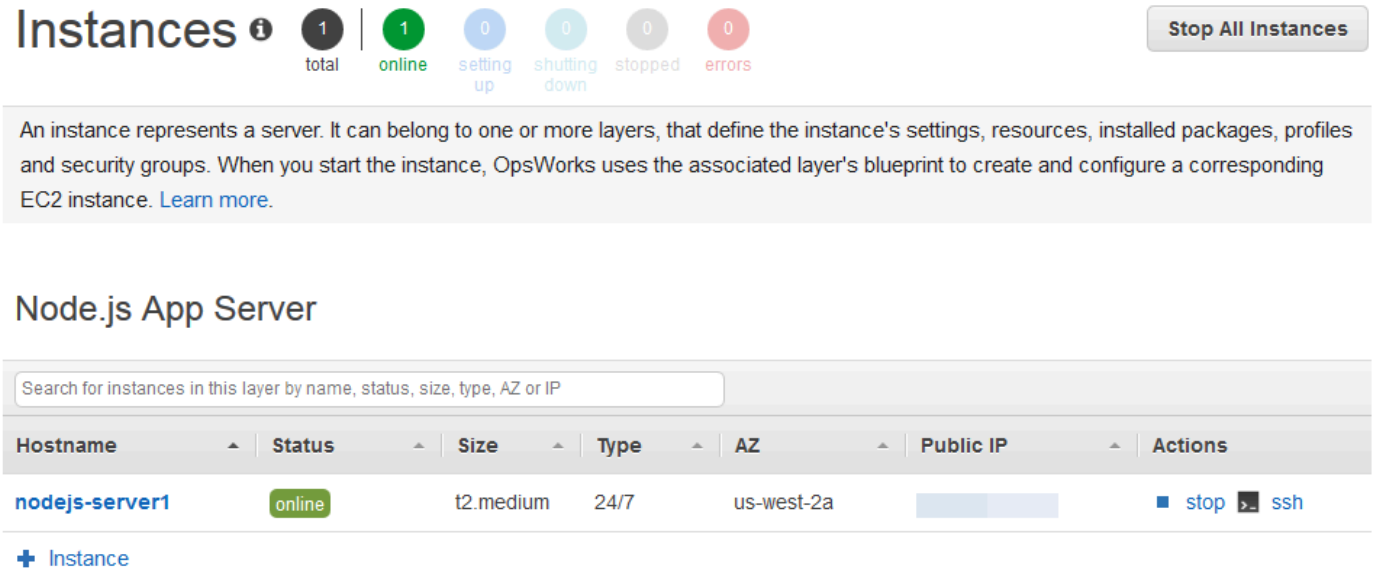
Étape 4 : Tester l'application déployée sur l'instance

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Testez les résultats du déploiement de l'application sur l'instance.

Pour tester le déploiement sur l'instance

1. Avec la page Instances affichée depuis l'étape précédente, pour Node.js App Server, pour nodejs-server1, pour Public IP (Adresse IP publique), choisissez l'adresse IP.



2. Sur la page web de félicitations, dans la zone de texte Leave a comment (Laisser un commentaire), entrez un commentaire, puis choisissez Send (Envoyer) pour tester l'application. L'application ajoute votre commentaire à la page web. Continuez à laisser des commentaires et à choisir Send (Envoyer) aussi souvent que vous le souhaitez.



3. Si vous avez un compte Twitter, choisissez Tweeter ou Suivre @ AWSOps Works, puis suivez les instructions qui s'affichent à l'écran pour tweeter à propos de l'application ou pour suivre @ AWSOps Works.

Vous avez testé avec succès l'application déployée sur l'instance.

Dans les étapes restantes, vous pouvez utiliser la console AWS OpsWorks Stacks pour explorer les paramètres de la pile et de ses composants. Au cours de l'étape suivante, vous pourrez commencer votre exploration par l'examen des paramètres de la pile.

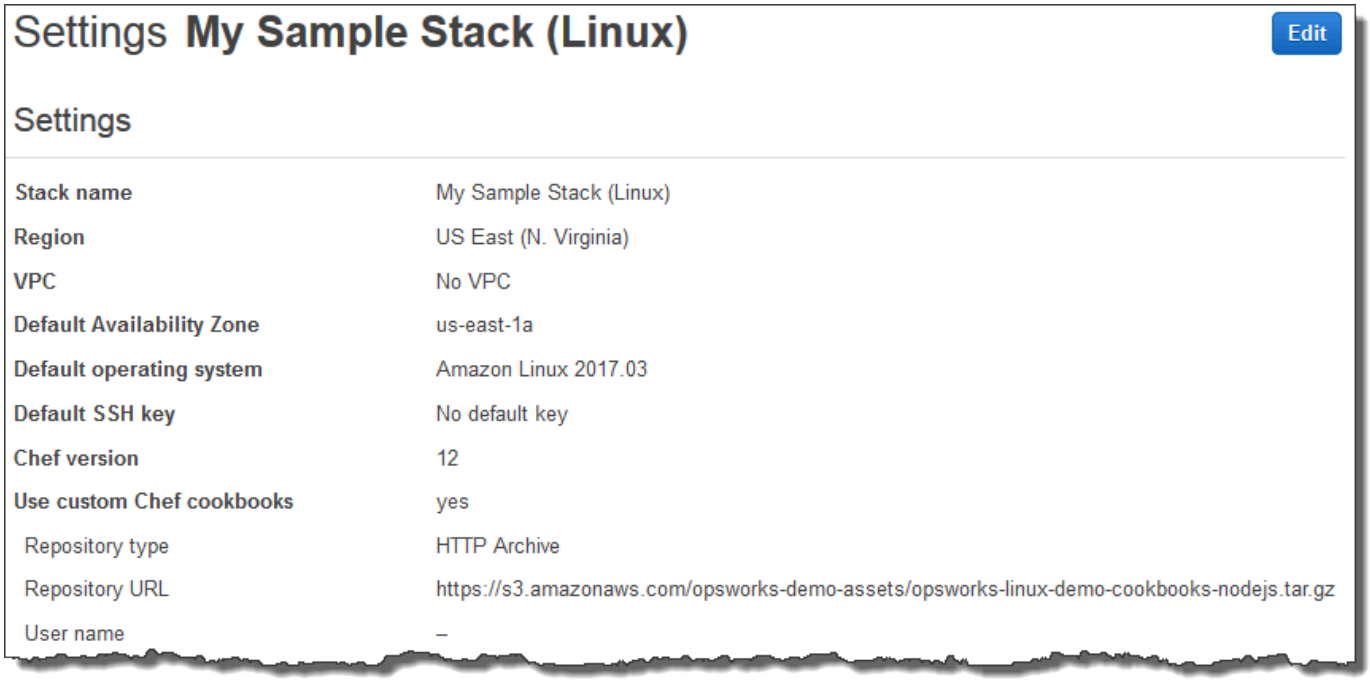
Étape 5 : Explorer les paramètres de la pile

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Examinez comment AWS OpsWorks Stacks configure la pile.

Pour afficher les paramètres de la pile

1. Dans la barre de navigation du service, choisissez Stack (Pile). La page My Sample Stack (Linux) [Mon exemple de pile (Linux)] s'affiche.
2. Choisissez Stack Settings (Paramètres de pile). La page Settings My Sample Stack (Linux) [Mon exemple de pile (Linux)] s'affiche :



Pour en savoir plus sur la plupart des paramètres, choisissez Edit (Modifier), puis passez votre souris au-dessus de chacun des paramètres. (Tous les paramètres n'ont pas une description à l'écran.) Pour plus d'informations sur ces paramètres, consultez la page Créer une pile.

Pour explorer le livre de recettes Chef utilisé dans cette procédure pas à pas, ouvrez le référentiel opsworks-linux-demo-cookbooks-nodejs sur. GitHub

Au cours de l'étape suivante, vous pourrez explorer les paramètres de la couche.

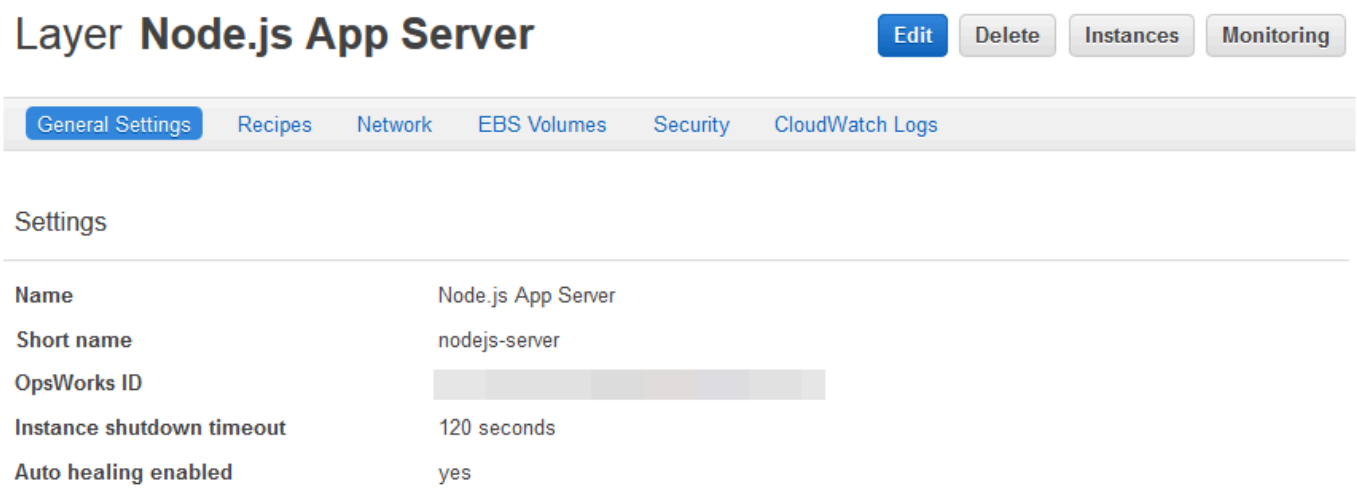
Étape 6 : Explorer les paramètres de la couche

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Examinez comment AWS OpsWorks Stacks configure la couche.

Pour afficher les paramètres de la couche

1. Dans le panneau de navigation du service, choisissez Layers (Couches). La page Layers (Couches) s'affiche.
2. Choisissez Node.js App Server (Serveur d'applications Node.js). La page Layer Node.js App Server (Couche Serveur d'applications Node.js) s'affiche. Pour afficher les paramètres de la couche, choisissez General Settings (Paramètres généraux), Recipes (Recettes), Network (Réseau), EBS Volumes (Volumes EBS) et Security (Sécurité) :



Pour en savoir plus sur la plupart des paramètres, choisissez Edit (Modifier), puis passez votre souris au-dessus de chacun des paramètres. (Tous les paramètres n'ont pas une description à l'écran.) Pour plus d'informations sur ces paramètres, consultez la page Modification de OpsWorks la configuration d'une couche.

Au cours de l'étape suivante, vous pourrez explorer les journaux et les paramètres de l'instance.

Étape 7 : Explorer les paramètres et journaux de l'instance

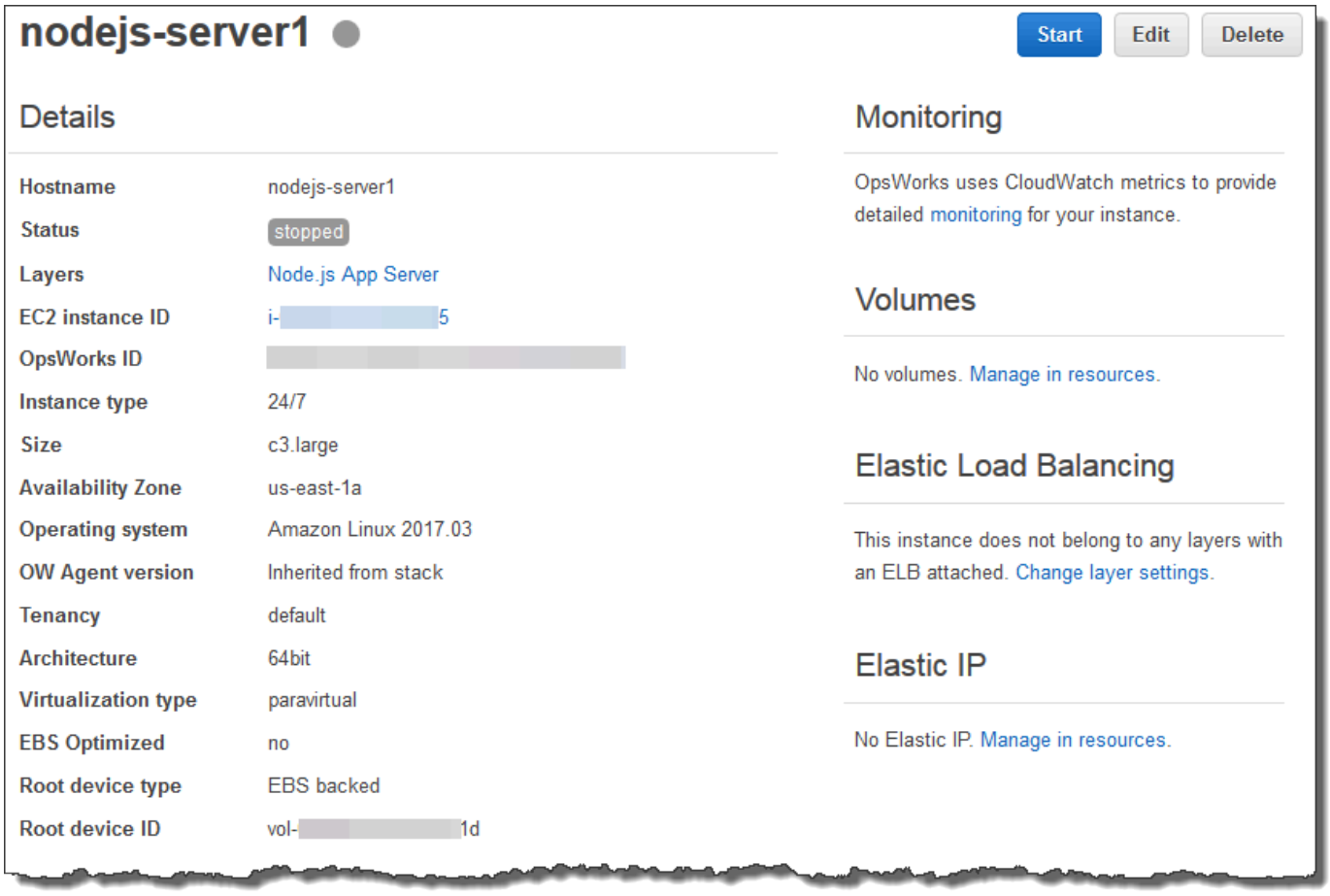
> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

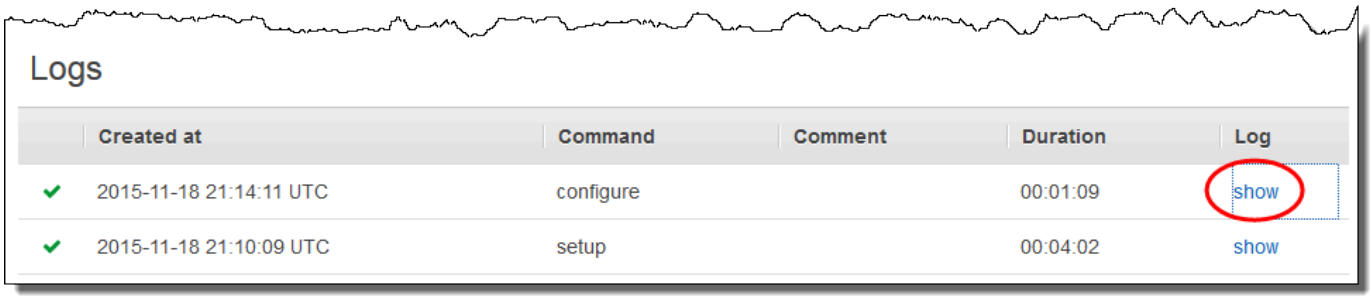
Examinez les paramètres utilisés par AWS OpsWorks Stacks pour lancer l'instance. Vous pouvez également examiner les journaux d'instance créés par AWS OpsWorks Stacks.

Pour afficher les paramètres et les journaux de l'instance

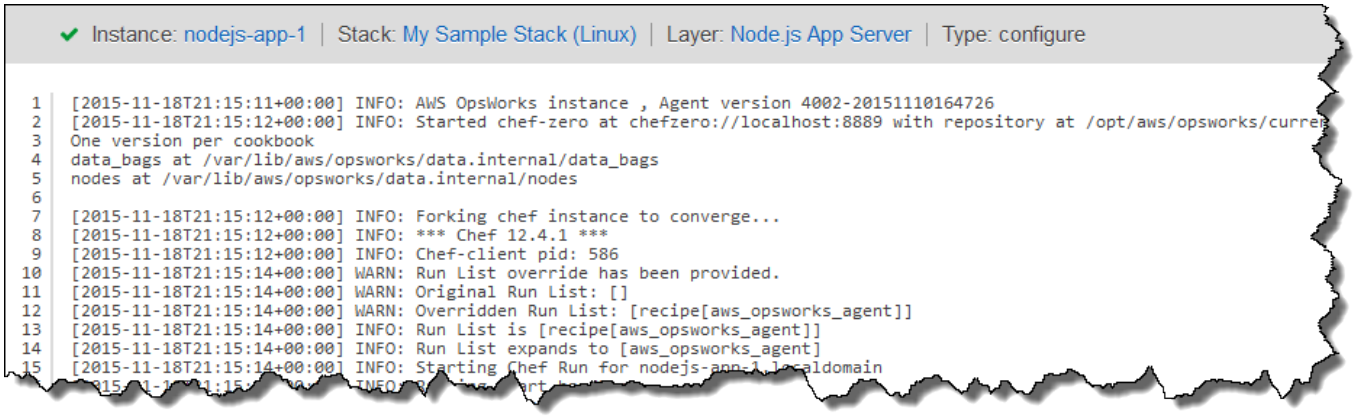
1. Dans le panneau de navigation du service, choisissez Instances. La page Instances s'affiche.
2. Pour Node.js App Server (Serveur d'applications Node.js), choisissez nodejs-server1. La page de propriétés de l'instance s'affiche.



3. Pour explorer les journaux de l'instance, dans la section Logs (Journaux), pour Log (Journal), choisissez show (afficher).



4. AWS OpsWorks Stacks affiche le journal dans un onglet de navigateur Web distinct.



Pour en savoir plus sur ce que certains paramètres d'instance représentent, revenez à la page nodejs-server1, choisissez Stop puis, quand vous voyez le message de confirmation, choisissez Stop. Choisissez Edit (Modifier) une fois que le Status (Statut) est passé de stopping (en cours d'arrêt) à stopped (arrêté), puis passez votre souris au-dessus de chacun des paramètres. (Tous les paramètres n'ont pas une description à l'écran.) Pour plus d'informations sur ces paramètres, consultez la page Ajout d'une instance à une couche.

Lorsque vous avez terminé la révision des paramètres, choisissez Start (Démarrer) pour redémarrer l'instance et patientez jusqu'à ce que Status (Statut) soit online (en ligne). Sinon, vous ne serez pas en mesure de tester l'application plus tard, parce que l'instance restera arrêtée.

> **Note**
>
> Si vous souhaitez vous connecter à l'instance pour l'explorer plus en profondeur, vous devez d'abord fournir à AWS OpsWorks Stacks des informations sur votre clé SSH publique (que vous pouvez créer à l'aide d'outils tels que ssh-keygen ou PuTTYgen), puis définir des

autorisations sur la pile My Sample Stack (Linux) pour permettre à votre utilisateur de se connecter à l'instance. Pour obtenir les instructions, consultez Enregistrement de la clé SSH publique d'un utilisateur et Connexion avec SSH.

Au cours de l'étape suivante, vous allez explorer les paramètres de l'application.

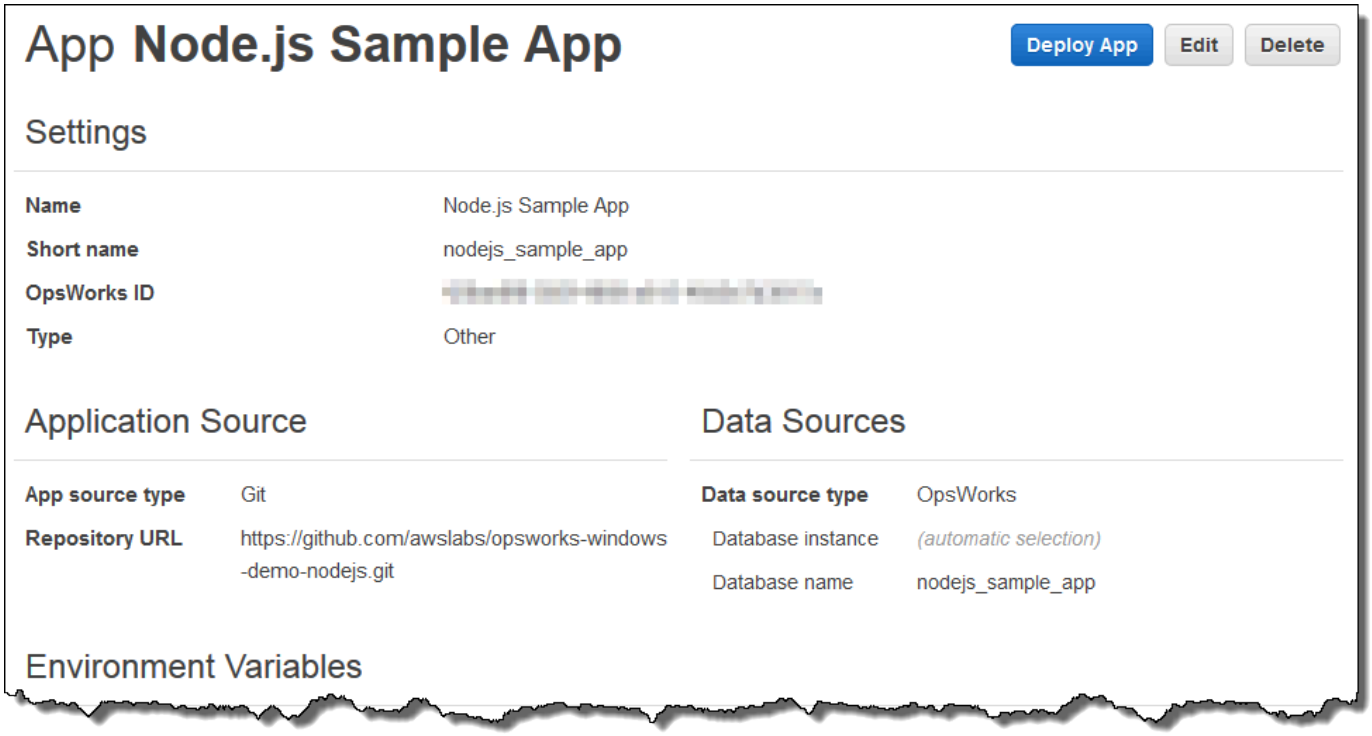
Étape 8 : Explorer les paramètres de l'application

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Examinez les paramètres utilisés par AWS OpsWorks Stacks pour l'application.

Pour afficher les paramètres de l'application

1. Dans le panneau de navigation du service, sélectionnez Apps (Applications). La page Apps (Applications) s'affiche.
2. Choisissez Node.js Sample App (Exemple d'application Node.js). La page App Node.js Sample App (Application Exemple d'application Node.js) s'affiche :



Pour en savoir plus sur ce que certains paramètres représentent, choisissez Edit (Modifier), puis passez votre souris au-dessus de chacun des paramètres. (Tous les paramètres n'ont pas une description à l'écran.) Pour plus d'informations sur ces paramètres, consultez Ajout d'applications.

Au cours de l'étape suivante, vous pourrez explorer les rapports d'analyse de la couche.

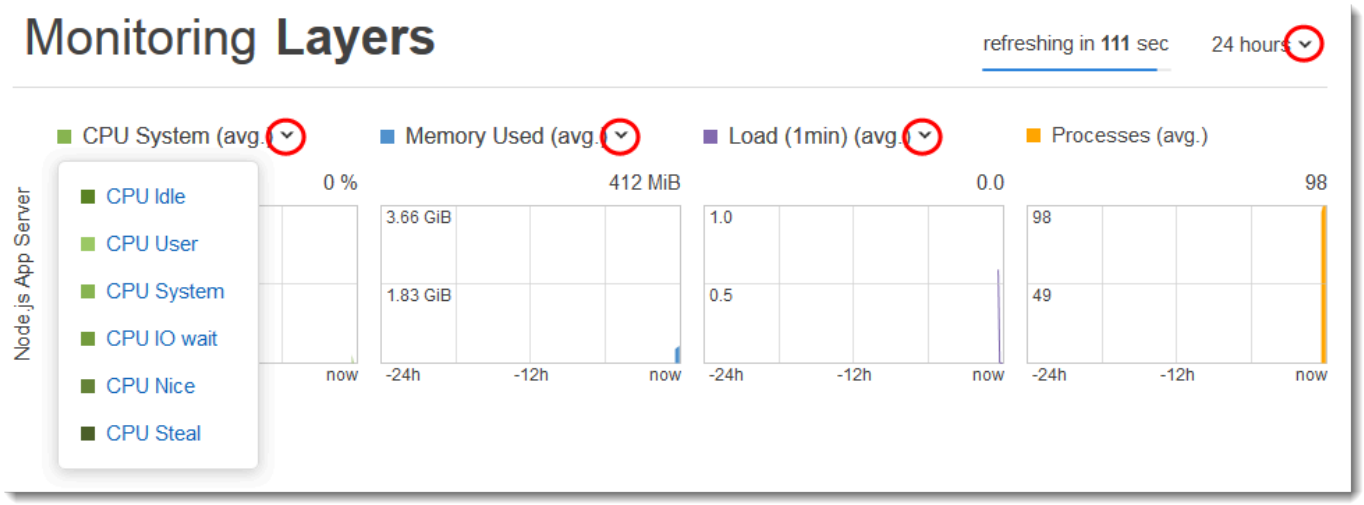
Étape 9 : Explorer les rapports de surveillance de la couche

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Examinez les rapports générés par AWS OpsWorks Stacks concernant les performances de calcul de la couche.

Pour afficher les rapports de surveillance de la couche

1. Dans le panneau de navigation du service, choisissez Monitoring (Surveillance). La page Monitoring Layers (Surveillance de couches) s'affiche.
2. Pour découvrir des vues supplémentaires, sélectionnez la flèche en regard de CPU (UC), Memory (Mémoire), Load (Charge) et le temps :



Pour plus d'informations sur ces rapports et sur d'autres rapports, consultez Utilisation d'Amazon CloudWatch et Surveillance.

Au cours de l'étape suivante, vous pourrez explorer les paramètres de pile supplémentaires.

Étape 10 : Explorer les paramètres supplémentaires de la pile

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Au cours de cette étape, vous pouvez examiner les paramètres supplémentaires de la pile.

AWS OpsWorks Stacks n'a effectué aucun déploiement distinct, n'a fourni aucune ressource supplémentaire et n'a ajusté aucune autorisation supplémentaire dans le cadre de cette pile.

Les pages Déploiements et Commandes, Ressources et Autorisations ne présentent donc pas beaucoup d'intérêt. Si vous souhaitez tout de même voir ces paramètres, choisissez Deployments (Déploiements), Resources (Ressources) et Permissions (Autorisations) dans le panneau de navigation de service, respectivement. Si vous souhaitez plus d'informations sur ce que représentent ces pages, consultez Déploiement d'applications, Gestion des ressources et Gestion des autorisations utilisateur.

À l'étape suivante, vous pouvez nettoyer les AWS ressources que vous avez utilisées pour cette procédure pas à pas. Cette étape suivante est facultative. Vous souhaiterez peut-être continuer à utiliser ces AWS ressources pour en savoir plus sur AWS OpsWorks Stacks. Cependant, la conservation de ces AWS ressources peut entraîner des frais permanents sur votre AWS compte. Si vous souhaitez conserver ces AWS ressources pour une utilisation ultérieure, vous avez maintenant terminé cette procédure pas à pas et vous pouvez passer àÉtapes suivantes.

Étape 11 : (Facultatif) Nettoyer

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Pour éviter d'encourir des frais supplémentaires sur votre AWS compte, vous pouvez supprimer l'application et les AWS ressources utilisées pour cette procédure pas à pas, y compris l'instance et la AWS OpsWorks pile Stacks. (Pour plus d'informations, consultez la section AWS OpsWorks Tarification.) Cependant, vous souhaiterez peut-être continuer à utiliser ces AWS ressources tout en continuant à en apprendre davantage sur AWS OpsWorks Stacks. Si vous souhaitez que ces AWS ressources restent disponibles, vous avez maintenant terminé cette procédure pas à pas et vous pouvez passer àÉtapes suivantes.

Le contenu stocké dans les ressources que vous avez créées pour cette procédure pas à pas peut contenir des informations d'identification personnelle. Si vous ne voulez plus que ces informations soient stockées par AWS, suivez les étapes décrites dans cette rubrique.

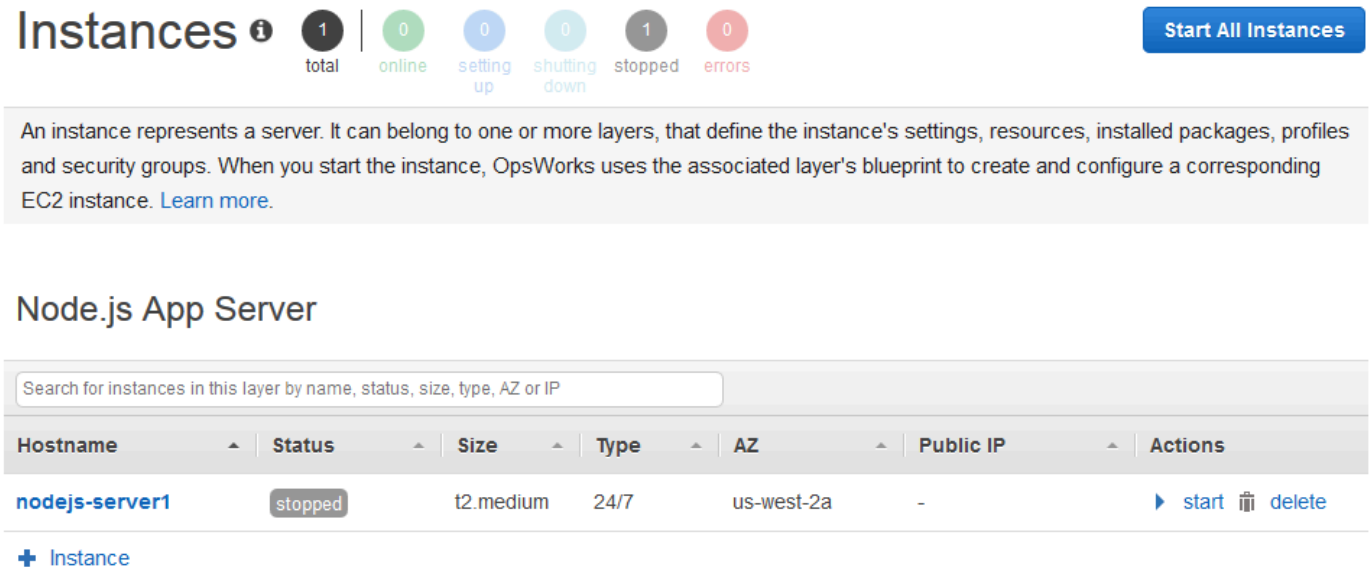
Pour supprimer l'application de la pile

1. Dans le panneau de navigation du service, sélectionnez Apps (Applications). La page Apps (Applications) s'affiche.
2. Pour Node.js Sample App, pour Actions, choisissez delete (supprimer). Dans le message de confirmation, sélectionnez Delete (Supprimer). Une fois l'application supprimée, vous voyez le message No apps (Aucune application).

Pour supprimer l'instance de la pile

1. Dans le panneau de navigation du service, choisissez Instances. La page Instances s'affiche.
2. Pour Node.js Application Server, pour nodejs-server1, pour Actions, choisissez stop. Dans le message de confirmation, sélectionnez Stop.

   Ce processus peut prendre quelques minutes. Lorsque AWS OpsWorks Stacks est terminé, les résultats suivants sont affichés.



3. Pour Actions, choisissez delete (supprimer). Dans le message de confirmation, sélectionnez Delete (Supprimer). L'instance a été supprimée et le message No instances (Aucune instance) s'affiche.

Pour supprimer la pile

1. Dans le panneau de navigation du service, choisissez Stack (Pile). La page My Sample Stack (Linux) [Mon exemple de pile (Linux)] s'affiche.
2. Choisissez Supprimer pile. Dans le message de confirmation, sélectionnez Delete (Supprimer). La pile est supprimée et la page du OpsWorkstableau de bord s'affiche.

Vous pouvez éventuellement supprimer l'utilisateur et la paire de EC2 clés Amazon que vous avez utilisés pour cette procédure pas à pas, si vous ne souhaitez pas les réutiliser pour accéder à d'autres AWS services et EC2 instances. Pour obtenir des instructions, consultez Supprimer un utilisateur IAM, des paires de EC2 clés Amazon et des instances Linux.

Vous avez maintenant terminé cette procédure pas à pas. Pour de plus amples informations, veuillez consulter Étapes suivantes.

Étapes suivantes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Maintenant que vous avez terminé cette procédure pas à pas, vous pouvez en savoir plus sur l'utilisation de AWS OpsWorks Stacks :

- Entraînez-vous à recréer manuellement cette pile vous-même à l'aide de AWS OpsWorks Stacks. Consultez Mise en route : Linux.
- Explorez le livre de recettes et l'application utilisés par AWS OpsWorks Stacks pour cette procédure pas à pas. Consultez En savoir plus : Explorer le livre de recettes utilisé dans cette procédure pas à pas et En savoir plus : Explorer l'application utilisée dans la procédure pas à pas dans la procédure Mise en route : Linux d'accompagnement.
- Entraînez-vous à utiliser AWS OpsWorks Stacks avec des instances Windows. Consultez Mise en route : Windows.

- Découvrez-en plus sur les piles en apprenant comment Créer une pile.
- Découvrez-en plus sur les couches en Modification de OpsWorks la configuration d'une couche.
- Découvrez-en plus sur les instances en Ajout d'une instance à une couche.
- Découvrez-en plus sur les applications en Déploiement d'applications.
- En savoir plus sur Livres de recettes et recettes.
- Créez vos propres livres de recettes. Consultez Mise en route : livres de recettes.
- Découvrez-en plus sur le contrôle de l'accès aux piles avec Sécurité et autorisations.

Mise en route avec les piles Linux

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Dans cette présentation, vous allez apprendre à utiliser AWS OpsWorks Stacks pour créer un environnement d'application Node.js. Lorsque vous aurez terminé, vous disposerez d'une instance Amazon Elastic Compute Cloud (Amazon EC2) exécutant Chef 12, d'un serveur HTTP Node.js et d'une application Web que vous pourrez utiliser pour interagir avec Twitter et laisser des commentaires sur une page Web.

Chef est un framework tiers destiné à configurer et à gérer des serveurs, tels que des EC2 instances, et à la manière dont les applications sont déployées et maintenues sur ces serveurs. Si vous ne connaissez pas Chef, après avoir terminé cette procédure pas à pas, nous vous recommandons d'en savoir plus sur Chef afin de profiter pleinement de tout ce que AWS OpsWorks Stacks a à offrir. (Pour plus d'informations, consultez le site web Découvrir Chef.)

AWS OpsWorks Stacks prend en charge quatre distributions Linux : Amazon Linux, Ubuntu Server, CentOS et Red Hat Enterprise Linux. Pour cette procédure pas à pas, nous utilisons Ubuntu Server. AWS OpsWorks Stacks fonctionne également avec Windows Server. Bien que nous disposions d'une procédure pas à pas équivalente pour les piles Windows Server, nous vous recommandons de commencer par la suivre afin d'apprendre les concepts de base relatifs à AWS OpsWorks Stacks et

Chef qui n'y sont pas repris. Une fois que vous avez terminé cette procédure pas à pas, consultez la procédure pas à pas Mise en route : Windows.

Rubriques

- Étape 1 : Exécuter les opérations prérequises
- Étape 2 : Créer une pile
- Étape 3 : Ajouter une couche à la pile
- Étape 4 : Spécifier l'application à déployer sur l'instance
- Étape 5 : Lancement d'une instance
- Étape 6 : Déployer l'application sur l'instance
- Étape 7 : Tester l'application déployée sur l'instance
- Étape 8 : (Facultatif) Nettoyer
- Étapes suivantes
- En savoir plus : Explorer le livre de recettes utilisé dans cette procédure pas à pas
- En savoir plus : Explorer l'application utilisée dans la procédure pas à pas

Étape 1 : Exécuter les opérations prérequises

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Complétez les étapes de configuration suivantes avant de commencer la procédure pas à pas. Ces étapes de configuration incluent la création d'un AWS compte, la création d'un utilisateur administratif et l'attribution d'autorisations d'accès à AWS OpsWorks Stacks.

Si vous avez déjà terminé la procédure pas à pas Mise en route : Exemple, vous remplissez les prérequis de cette procédure et pouvez passer directement à Étape 2 : Créer une pile.

Rubriques

- Inscrivez-vous pour un Compte AWS
- Création d'un utilisateur doté d'un accès administratif
- Attribuer des autorisations d'accès au service

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

   Une partie de la procédure d'inscription consiste à recevoir un appel téléphonique ou un message texte et à saisir un code de vérification sur le clavier du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à https://aws.amazon.com/et en choisissant Mon compte.

Création d'un utilisateur doté d'un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez Utilisateur racine d'un compte AWS AWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez Connexion en tant qu'utilisateur racine dans le Guide de l'utilisateur Connexion à AWS .

2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, voir Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d' AWS IAM Identity Center dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu'utilisateur doté d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

  Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribution d'un accès à d'autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d'autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

   Pour obtenir des instructions, consultez Création d'un ensemble d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l'utilisateur AWS IAM Identity Center .

Attribuer des autorisations d'accès au service

Activez l'accès au service AWS OpsWorks Stacks (et aux services connexes sur lesquels AWS OpsWorks Stacks s'appuie) en ajoutant les `AmazonS3FullAccess` autorisations `AWSOpsWorks_FullAccess` et à votre rôle ou à votre utilisateur.

Pour plus d'informations sur l'ajout d'autorisations, consultez la section Ajout d'autorisations d'identité IAM (console).

Vous avez maintenant terminé toutes les étapes de configuration et pouvez démarrer cette procédure pas à pas.

Étape 2 : Créer une pile

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous allez utiliser la console AWS OpsWorks Stacks pour créer une pile. Une pile est un ensemble d'instances et de AWS ressources associées ayant un objectif commun et que vous souhaitez gérer ensemble. (Pour plus d'informations, consultez Piles.) Pour cette procédure pas à pas, il n'y a qu'une seule instance.

Avant de commencer, remplissez les prérequis, si vous ne l'avez déjà fait.

Pour créer la pile

1. Connectez-vous à la AWS OpsWorks console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/opsworks/.
2. Exécutez l'une des actions suivantes, si elles s'appliquent :

- Si la page Bienvenue dans AWS OpsWorks Stacks s'affiche, choisissez Ajouter votre première pile ou Ajouter votre première pile AWS OpsWorks Stacks (les deux choix ont le même effet). La page Add stack (Ajouter une pile) s'affiche.
- Si la page OpsWorks Tableau de bord s'affiche, choisissez Ajouter une pile. La page Add stack (Ajouter une pile) s'affiche.

3. Avec la page Add stack (Ajouter une pile) affichée, choisissez Chef 12 stack (Pile Chef 12) si ce n'est déjà fait.
4. Dans la zone Stack name (Nom de la pile), entrez un nom, par exemple **`MyLinuxDemoStack`**. (Vous pouvez saisir un autre nom, mais n'oubliez pas de remplacer `MyLinuxDemoStack` tout au long de la procédure pas à pas.)
5. Pour Région, choisissez USA West (Oregon).
6. Pour VPC, effectuez l'une des actions suivantes :
   - Si un VPC est disponible, choisissez-le. (Pour plus d'informations, consultez Running a Stack in a VPC.)
   - Sinon, choisissez No VPC.
7. Pour Default operating system (Système d'exploitation par défaut), choisissez Linux et Ubuntu 18.04 LTS.
8. Pour Use custom Chef cookbooks (Utiliser les livres de recettes Chef personnalisés), choisissez Yes (Oui).
9. Pour Repository type (Type de référentiel), choisissez Http Archive (Archive HTTP).
10. Pour Repository URL (URL du référentiel), tapez **`https://s3.amazonaws.com/opsworks-demo-assets/opsworks-linux-demo-cookbooks-nodejs.tar.gz`**
11. Conservez les valeurs par défaut pour les éléments suivants :
    - Default Availability Zone (Zone de disponibilité par défaut) (us-west-2a)
    - Default SSH key (Clé SSH par défaut) (Do not use a default SSH key [Ne pas utiliser de clé SSH par défaut])
    - User name (Nom d'utilisateur) (vide)
    - Password (Mot de passe) (vide)
    - Stack color (Couleur de la pile) (bleu foncé)
12. Choisir Advanced (Avancé).

13. Pour le rôle IAM, effectuez l'une des opérations suivantes (pour plus d'informations, voirPermettre à AWS OpsWorks Stacks d'agir en votre nom) :

    - S'il aws-opsworks-service-roleest disponible, choisissez-le.
    - Si aws-opsworks-service-rolece n'est pas le cas, choisissez Nouveau rôle IAM.

14. Pour le profil d'instance IAM par défaut, effectuez l'une des opérations suivantes (pour plus d'informations, consultezSpécification des autorisations pour les applications exécutées sur EC2 des instances) :

    - Si aws-opsworks-ecdeux rôles sont disponibles, choisissez-le.
    - Si aws-opsworks-ec2 rôles n'est pas disponible, choisissez Nouveau profil d'instance IAM.

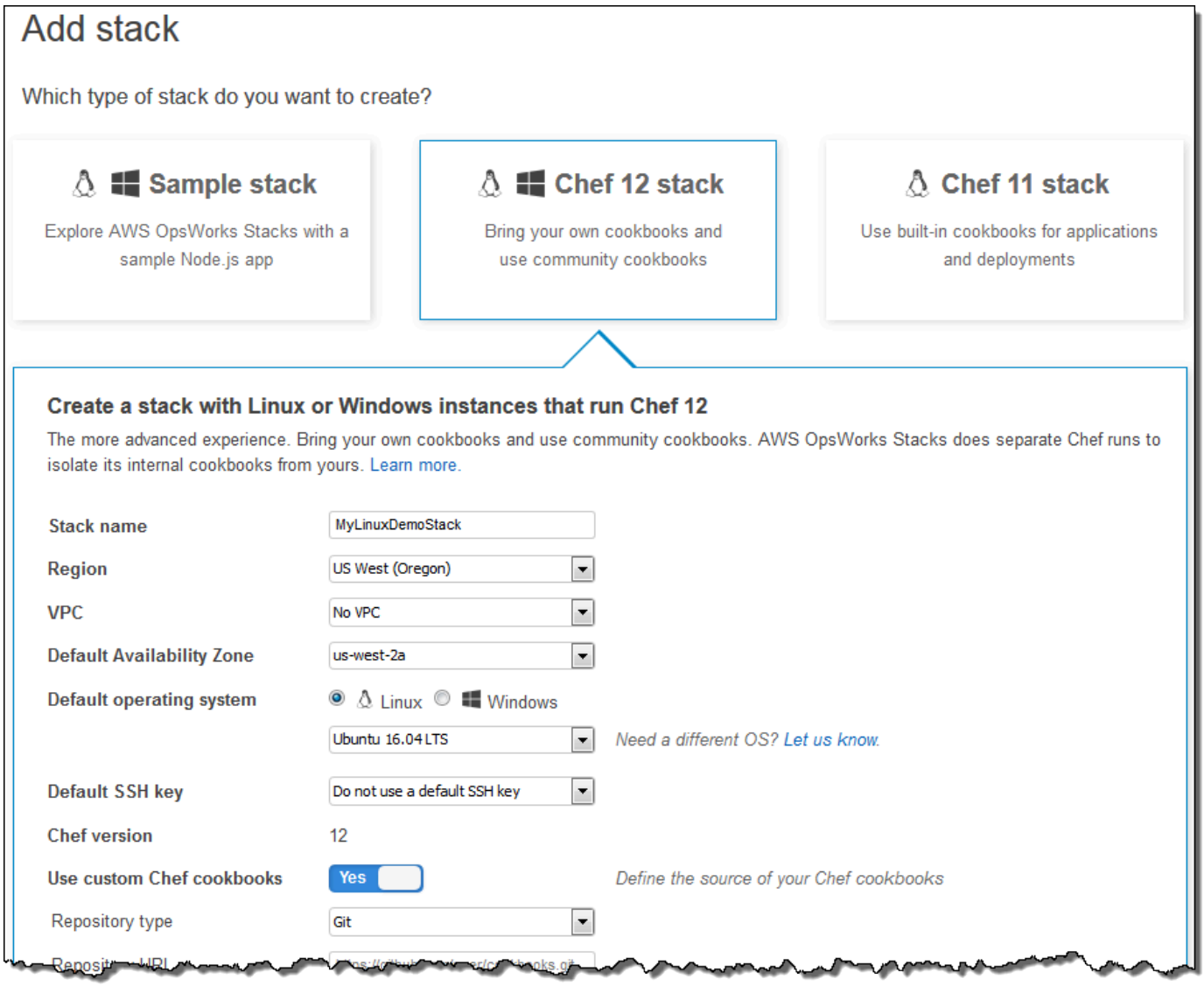
15. Pour API endpoint region (Région du point de terminaison de l'API), choisissez le point de terminaison régional de l'API avec lequel vous souhaitez associer la pile. Si vous souhaitez que la pile se trouve dans la région de l'ouest des États-Unis (Oregon) au sein du point de terminaison régional des États-Unis est (Virginie du Nord), choisissez us-east-1. Si vous souhaitez que la pile se trouve à la fois dans la région USA Ouest (Oregon) et associée au point de terminaison régional USA Ouest (Oregon), choisissez us-west-2.

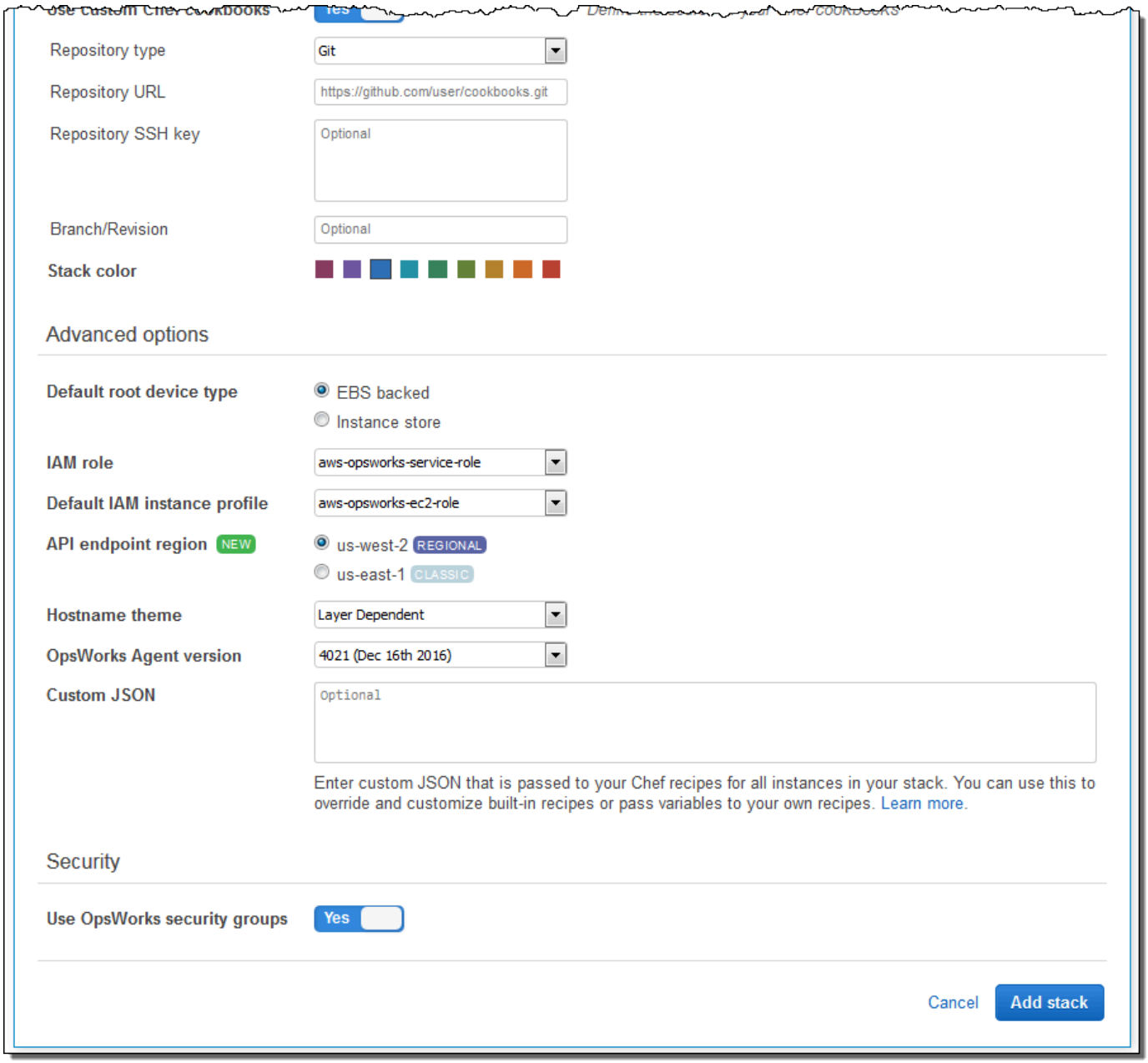
    > **Note**
    >
    > Le point de terminaison régional de l'est des États-Unis (Virginie du Nord) inclut Régions AWS les plus anciens pour des raisons de rétrocompatibilité, mais il est recommandé de choisir le point de terminaison régional le plus proche de celui que vous gérez AWS. Pour de plus amples informations, veuillez consulter Prise en charge de la région.

16. Conservez les valeurs par défaut pour les éléments suivants :

    - Default root device type (Type du périphérique racine par défaut) (EBS backed [Basé sur EBS])
    - Hostname theme (Thème du nom d'hôte) (Layer Dependent [Dépendant de la couche])
    - OpsWorks Version de l'agent (version la plus récente)
    - Custom JSON (JSON personnalisé) (vide)
    - Utiliser des groupes OpsWorks de sécurité (Oui)

17. Vos résultats doivent correspondre aux captures d'écran suivantes, à l'exception peut-être du VPC, du rôle IAM et du profil d'instance IAM par défaut :


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. Choisissez Add Stack. AWS OpsWorks Stacks crée la pile et affiche la MyLinuxDemoStackpage.

Vous avez maintenant une pile avec les paramètres appropriés pour cette procédure pas à pas.

Dans l'étape suivante, vous allez ajouter une couche à la pile.

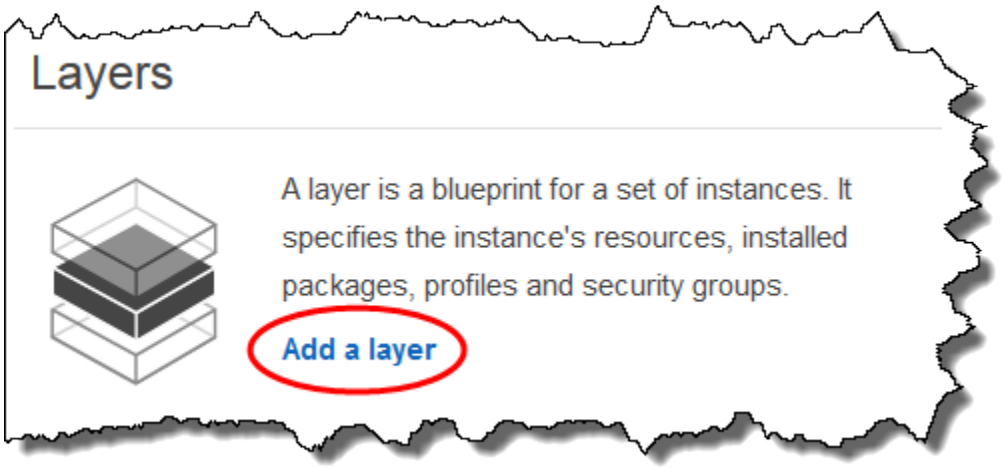
Étape 3 : Ajouter une couche à la pile

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

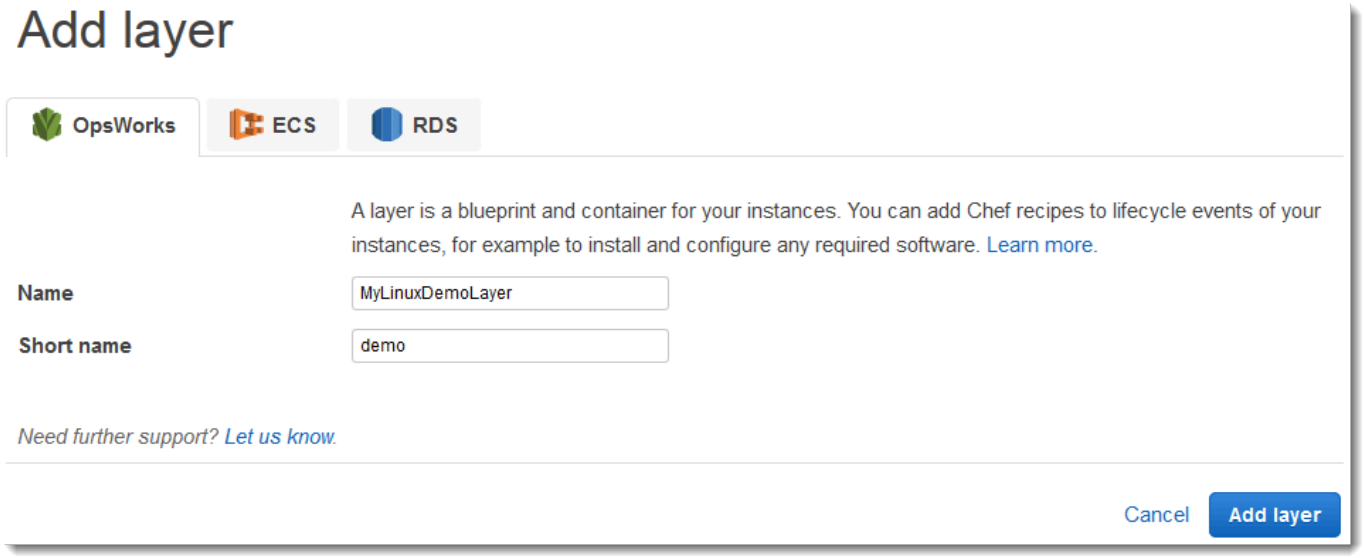
Une couche est un modèle pour un ensemble d'instances, telles que les EC2 instances Amazon. Elle spécifie des informations telles que les paramètres, les ressources, les packages installés et les groupes de sécurité de l'instance. Ensuite, ajoutez une couche à la pile. (Pour plus d'informations sur les couches, consultez Couches.)

Pour ajouter la couche à la pile

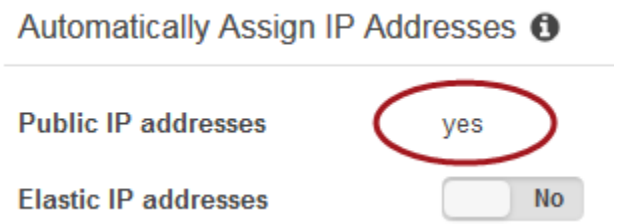
1. Avec la MyLinuxDemoStackpage affichée à l'étape précédente, pour Couches, choisissez Ajouter une couche :



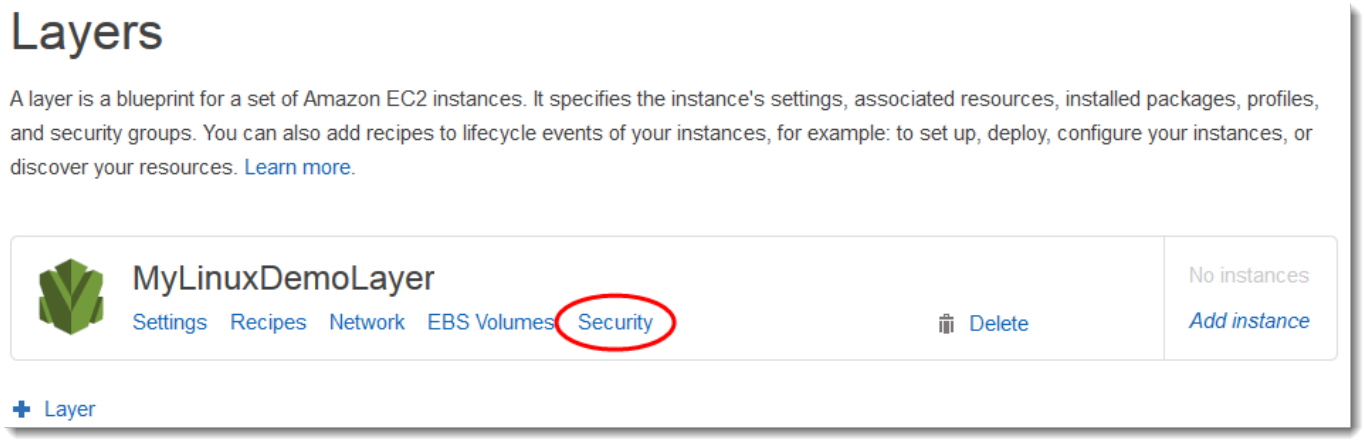
2. La page Add Layer (Ajouter une couche) s'affiche. OpsWorksDans l'onglet, pour Nom, tapez**`MyLinuxDemoLayer`**. (Vous pouvez saisir un autre nom, mais n'oubliez pas de remplacer `MyLinuxDemoLayer` tout au long de la procédure pas à pas.)
3. Pour Short name (Nom court), tapez **`demo`** (vous pouvez entrer une autre valeur, mais veillez à remplacer `demo` tout au long de la procédure) :



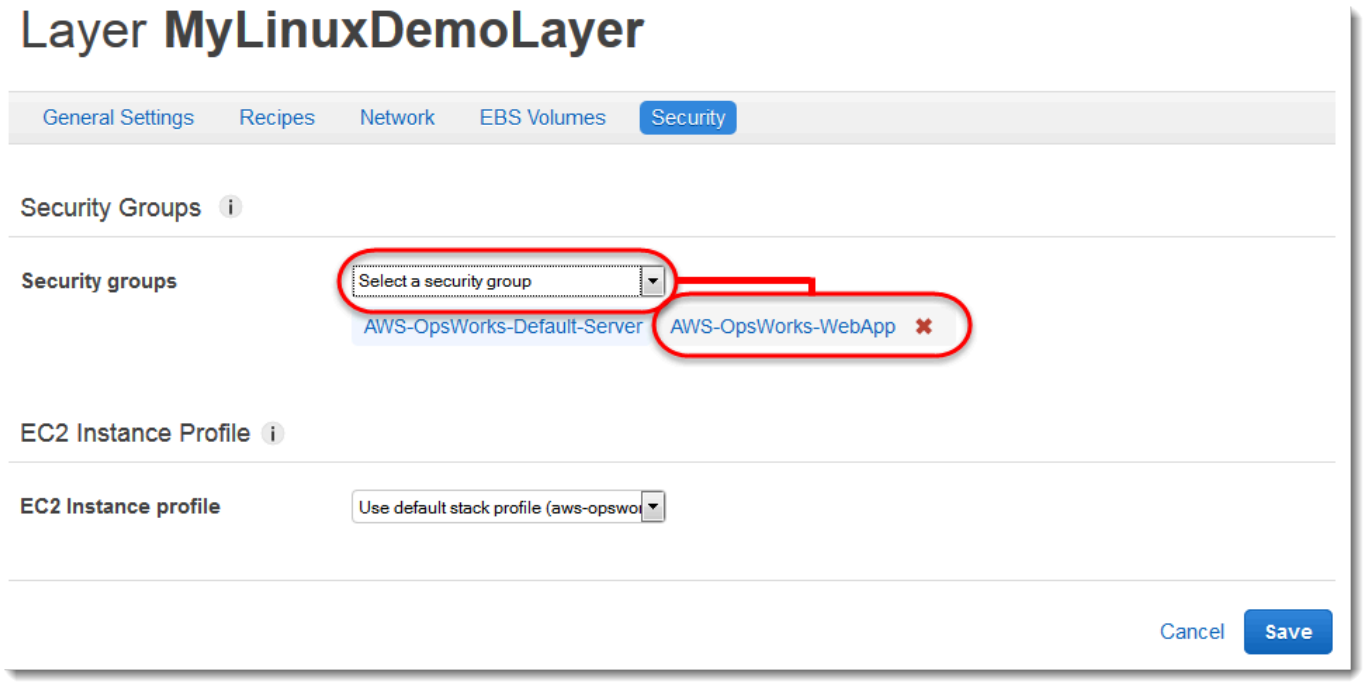
4. Choisissez Ajouter une couche. AWS OpsWorks Stacks crée la couche et affiche la page Couches.
5. Sur la page Couches, pour MyLinuxDemoLayer, sélectionnez Réseau.
6. Dans l'onglet Network (Réseau), sous Automatically Assign IP Addresses (Attribuer automatiquement les adresses IP), vérifiez que Public IP addresses (Adresses IP publiques) est configuré sur yes (oui). Si vous avez apporté des modifications, choisissez Save (Enregistrer).



7. Sur la page Layers (Couches), choisissez Security (Sécurité) :



8. La MyLinuxDemoLayer page Couche s'affiche avec l'onglet Sécurité ouvert. Pour les groupes de sécurité, choisissez AWS- OpsWorks - WebApp, puis cliquez sur Enregistrer :



9. Le groupe de sécurité `AWS-OpsWorks-WebApp` est ajouté à la couche. (Ce groupe de sécurité permet aux utilisateurs de se connecter à l'application sur l'instance ultérieurement au cours de cette procédure pas à pas. Sans ce groupe de sécurité, les utilisateurs recevront un message dans leur navigateur Web les informant qu'ils ne peuvent pas se connecter à l'instance.)

Vous avez maintenant une couche avec les paramètres appropriés pour cette procédure pas à pas.

Dans l'étape suivante, vous spécifiez l'application à déployer sur l'instance.

Étape 4 : Spécifier l'application à déployer sur l'instance

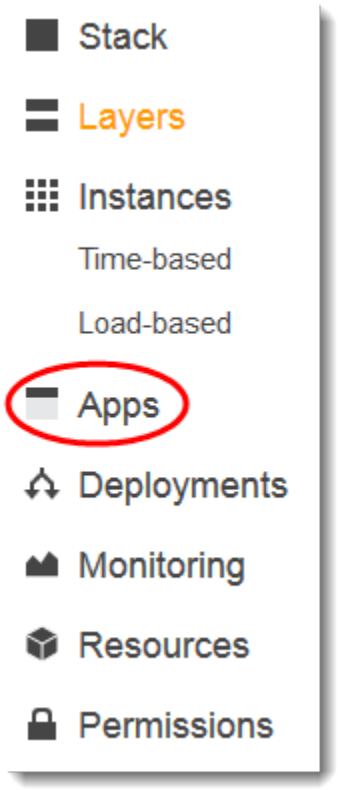
> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Indiquez à AWS OpsWorks Stacks l'application que vous déploierez sur l'instance ultérieurement au cours de cette procédure pas à pas. Dans ce contexte, AWS OpsWorks Stacks définit une application comme du code que vous souhaitez exécuter sur une instance. (Pour plus d'informations, consultez Applications.)

La procédure de cette section s'applique à Chef 12 et aux piles les plus récentes. Pour plus d'informations sur l'ajout d'applications aux couches des piles Chef 11, consultez Étape 2.4 : Créer et déployer une application - Chef 11.

Pour spécifier l'application à déployer

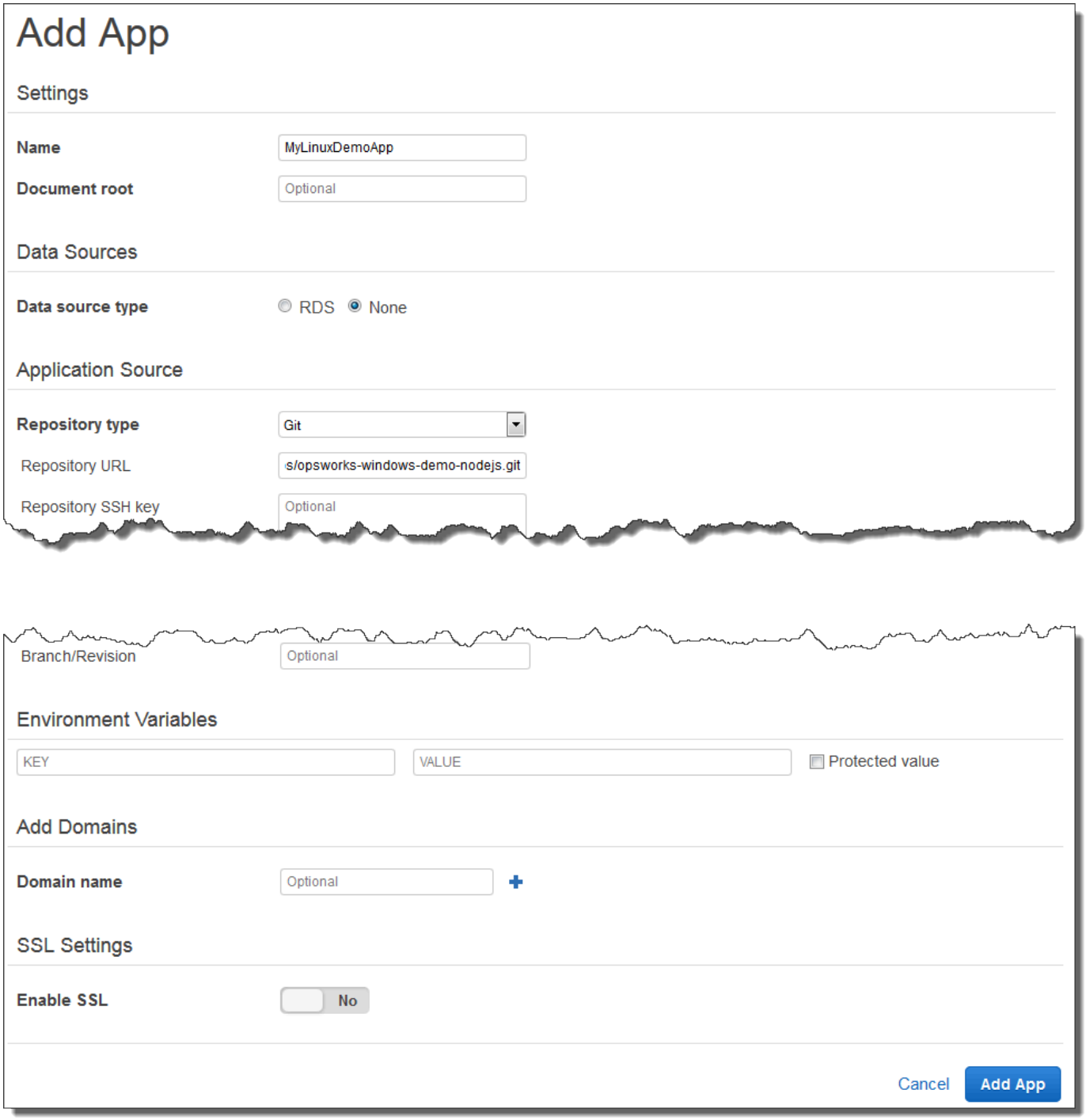
1. Dans le panneau de navigation du service, sélectionnez Apps (Applications) :



2. La page Apps (Applications) s'affiche. Choisissez Add an app (Ajouter une application). La page Add App (Ajouter une application) s'affiche.
3. Pour Settings (Paramètres), pour Name (Nom), tapez **`MyLinuxDemoApp`**. (Vous pouvez saisir un autre nom, mais n'oubliez pas de remplacer `MyLinuxDemoApp` tout au long de la procédure pas à pas.)
4. Pour Application Source (Source de l'application), pour Repository URL (URL du référentiel), tapez **`https://github.com/awslabs/opsworks-windows-demo-nodejs.git`**
5. Conservez les valeurs par défaut pour les éléments suivants :

   - Settings (Paramètres), Document root (Racine du document) (vide)
   - Data Sources (Sources de données), Data source type (Type de source de données) (None [Aucun])
   - Repository type (Type de référentiel) (Git)
   - Repository SSH key (Clé SSH du référentiel) (vide)
   - Branch/Revision (Branche/Révision) (vide)
   - Environment Variables (Variables d'environnement) (KEY vide, VALUE vide, Protected Value (Valeur protégée) non cochée)

- Add Domains (Ajouter un domaine), Domain Name (Nom de domaine) (vide)
- SSL Settings (Paramètres SSL), Enable SSL (Activer SSL) (No [Non])



6. Choisissez Ajouter une application. AWS OpsWorks Stacks ajoute l'application et affiche la page Applications.

Vous avez maintenant une application avec les paramètres appropriés pour cette procédure pas à pas.

Dans l'étape suivante, vous allez lancer l'instance.

Étape 5 : Lancement d'une instance

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

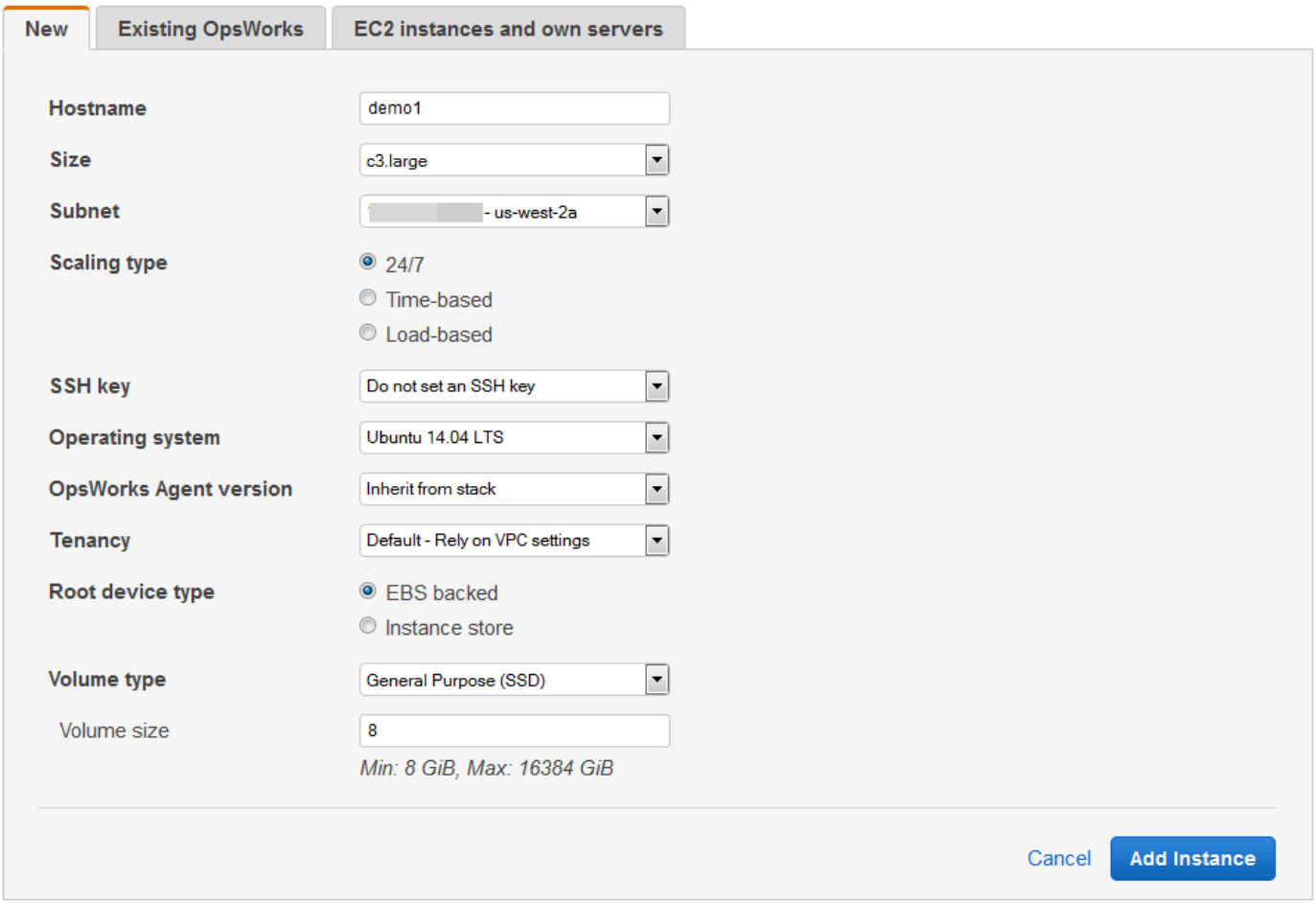
Utilisez AWS OpsWorks Stacks pour démarrer une EC2 instance Amazon du serveur Ubuntu. Cette instance utilise les paramètres que vous avez définis dans la couche créée précédemment dans la procédure pas à pas. (Pour plus d'informations, consultez instances.)

Pour lancer l'instance

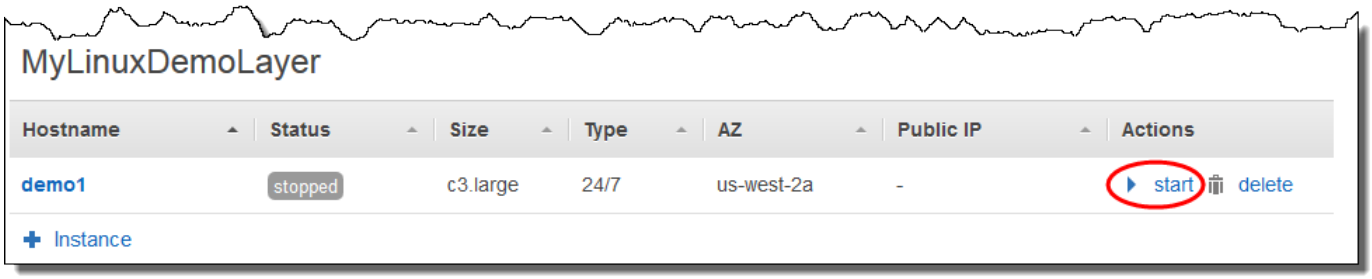
1. Dans le panneau de navigation du service, choisissez Instances. La page Instances s'affiche.
2. Pour MyLinuxDemoLayer, choisissez Ajouter une instance.
3. Sous l'onglet New (Nouveau), conservez les valeurs par défaut pour les éléments suivants :
   - Hostname (Nom d'hôte) (demo1)
   - Size (Taille) (c3.large)
   - Sous-réseau (*IP address*us-west-2a)
4. Choisir Advanced (Avancé).
5. Conservez les valeurs par défaut pour les éléments suivants :
   - Scaling type (Type de dimensionnement) (24/7)
   - SSH key (Clé SSH) (Do not use a default SSH key [Ne pas utiliser de clé SSH par défaut])
   - Operating system (Ubuntu 18.04 LTS)
   - OpsWorks Version de l'agent (hériter de la pile)

- Tenancy (Location) (Default - Rely on VPC settings [Par défaut - S'appuyer sur les paramètres VPC])
- Root device type (Type de périphérique racine) (EBS backed [Basé sur EBS])
- Volume type (Type de volume) (General Purpose (SSD) [Usage général (SSD)])
- Volume size (Taille du volume) (8)

6. Vos résultats seront similaires à capture d'écran suivante :



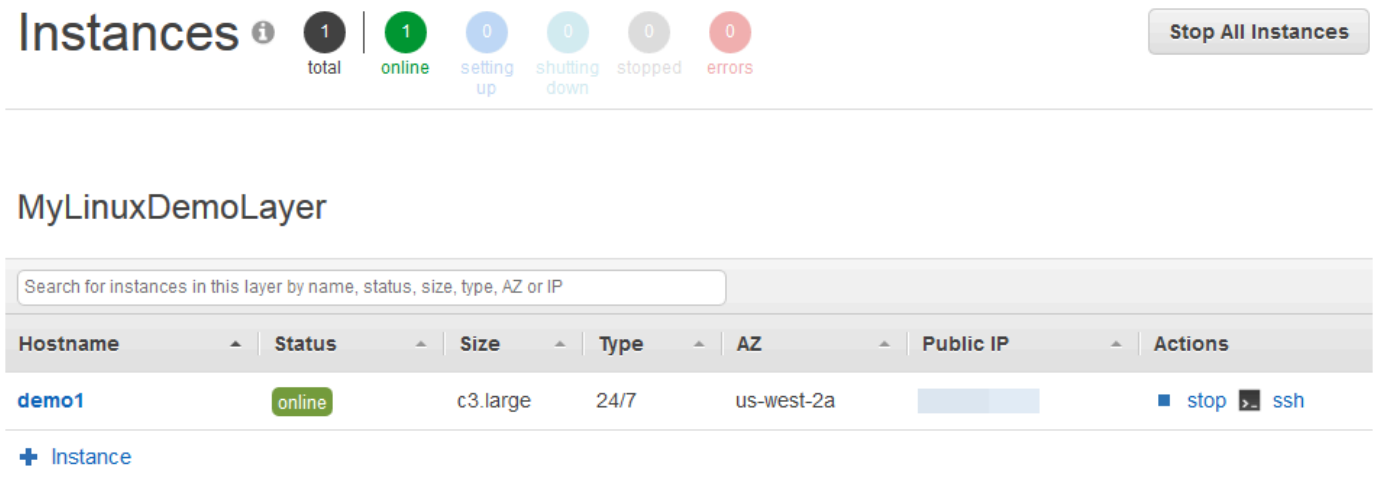
7. Choisissez Ajouter une instance. AWS OpsWorks Stacks ajoute l'instance à la couche et affiche la page Instances.
8. Pour MyLinuxDemoLayer, pour demo1, pour Actions, choisissez start :



9. Voici ce qui se produit alors :

   - Le cercle setting up (installation) passe de 0 à 1.
   - Status (Statut) passe successivement de stopped (arrêté) à requested (demandé), pending (en attente), booting (en cours de démarrage), running_setup (installation en cours) et, enfin, online (en ligne). Notez que ce processus peut prendre plusieurs minutes.
   - Après que Status (Statut) a pris la valeur online (en ligne), l'indicateur circulaire setting up (installation) passe de 1 à 0, et le cercle online (en ligne) passe de 0 à 1 et devient vert clair. Ne poursuivez pas tant que le cercle online (en ligne) n'est pas vert clair et que l'instance 1 en ligne n'est pas affichée.

10. Vos résultats doivent correspondre à la capture d'écran suivante avant que vous ne continuiez (si vous recevez un message d'erreur, vous pouvez consulter le Guide de débogage et dépannage) :



Vous avez maintenant une instance qui est prête pour y déployer l'application.

> **Note**
>
> Si vous souhaitez vous connecter à l'instance pour l'explorer plus en profondeur, vous devez d'abord fournir à AWS OpsWorks Stacks des informations sur votre clé SSH publique (que vous pouvez créer à l'aide d'outils tels que ssh-keygen ou PuTTYgen), puis vous devez définir des autorisations sur la `MyLinuxDemoStack` pile pour permettre à votre utilisateur de se connecter à l'instance. Pour obtenir les instructions, consultez Enregistrement de la clé SSH publique d'un utilisateur et Connexion avec SSH. Si vous envisagez d'utiliser SSH pour vous connecter à des instances via PuTTY, consultez la section Connexion à votre instance Linux depuis Windows à l'aide de PuTTY dans la documentation. AWS

Dans l'étape suivante, vous allez déployer l'application sur l'instance.

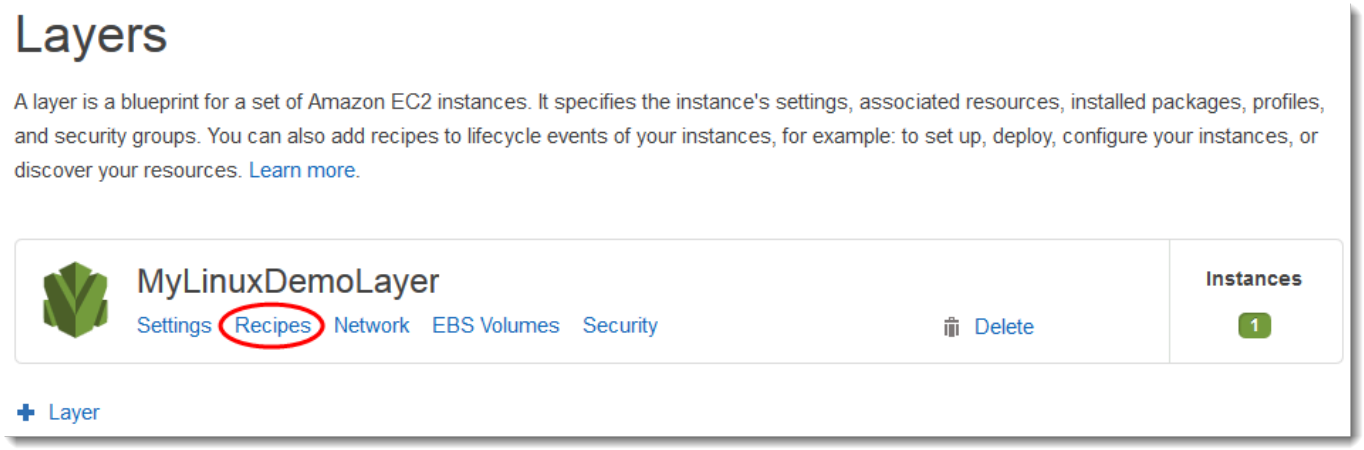
Étape 6 : Déployer l'application sur l'instance

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Au cours de cette étape, vous allez déployer l'application depuis GitHub l'instance en cours d'exécution. (Pour plus d'informations, consultez Déploiement d'applications.) Avant de déployer l'application, vous devez spécifier la recette à utiliser pour coordonner le déploiement. Une recette est un concept Chef. Les recettes sont des instructions, écrites avec la syntaxe du langage Ruby, qui spécifient les ressources à utiliser et l'ordre dans lequel ces ressources sont appliquées. (Pour plus d'informations, consultez À propos des recettes sur le site web Découvrir Chef.)

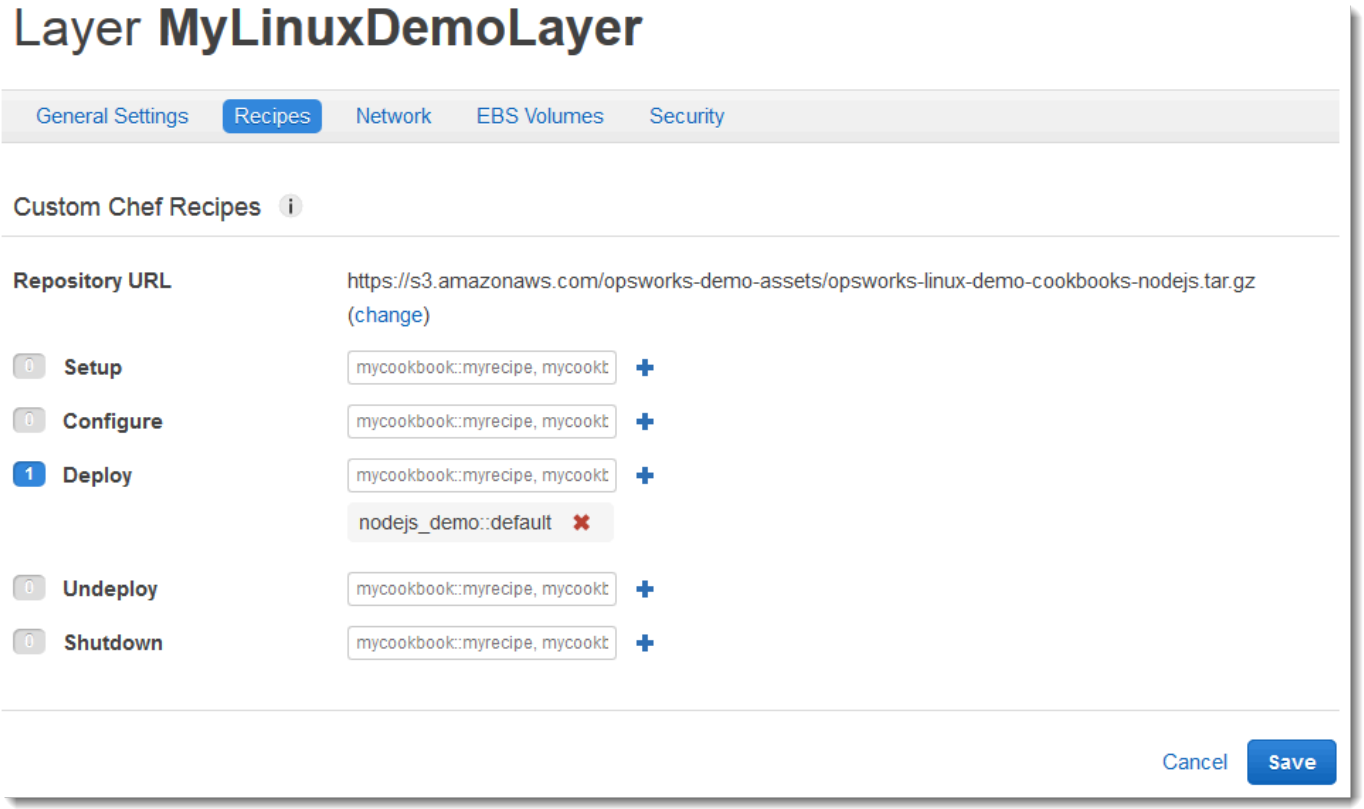
Pour spécifier la recette à utiliser pour déployer l'application sur l'instance

1. Dans le panneau de navigation du service, choisissez Layers (Couches). La page Layers (Couches) s'affiche.
2. Pour MyLinuxDemoLayer, choisissez Recipes :



La MyLinuxDemoLayer page Layer s'affiche avec l'onglet Recettes ouvert.

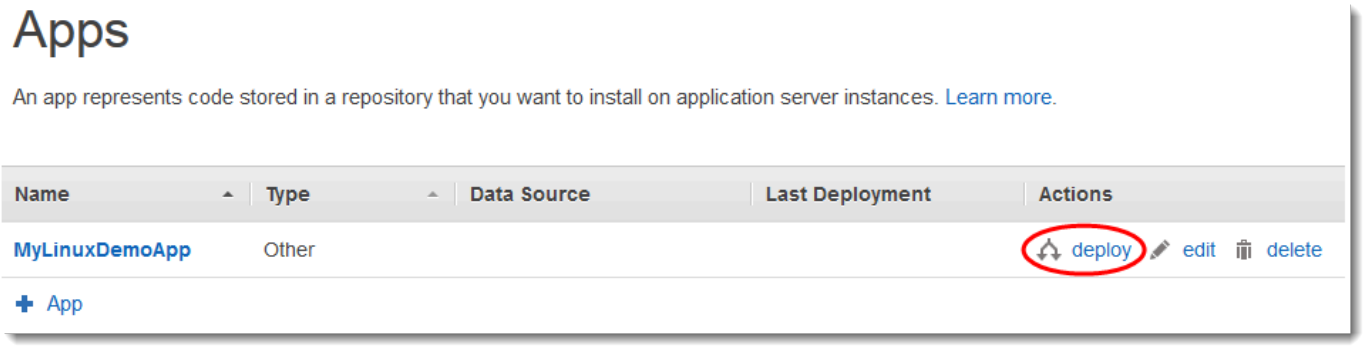
3. Pour Custom Chef Recipes (Recettes Chef personnalisées), pour Deploy (Déployer), tapez **`nodejs_demo::default`**, puis appuyez sur Entrée. `nodejs_demo` est le nom du livre de recettes et `default` est le nom de la recette cible du livre de recettes. (Pour explorer le code de la recette, consultez En savoir plus : Explorer le livre de recettes utilisé dans cette procédure pas à pas.) Vos résultats doivent correspondre à la capture d'écran suivante :



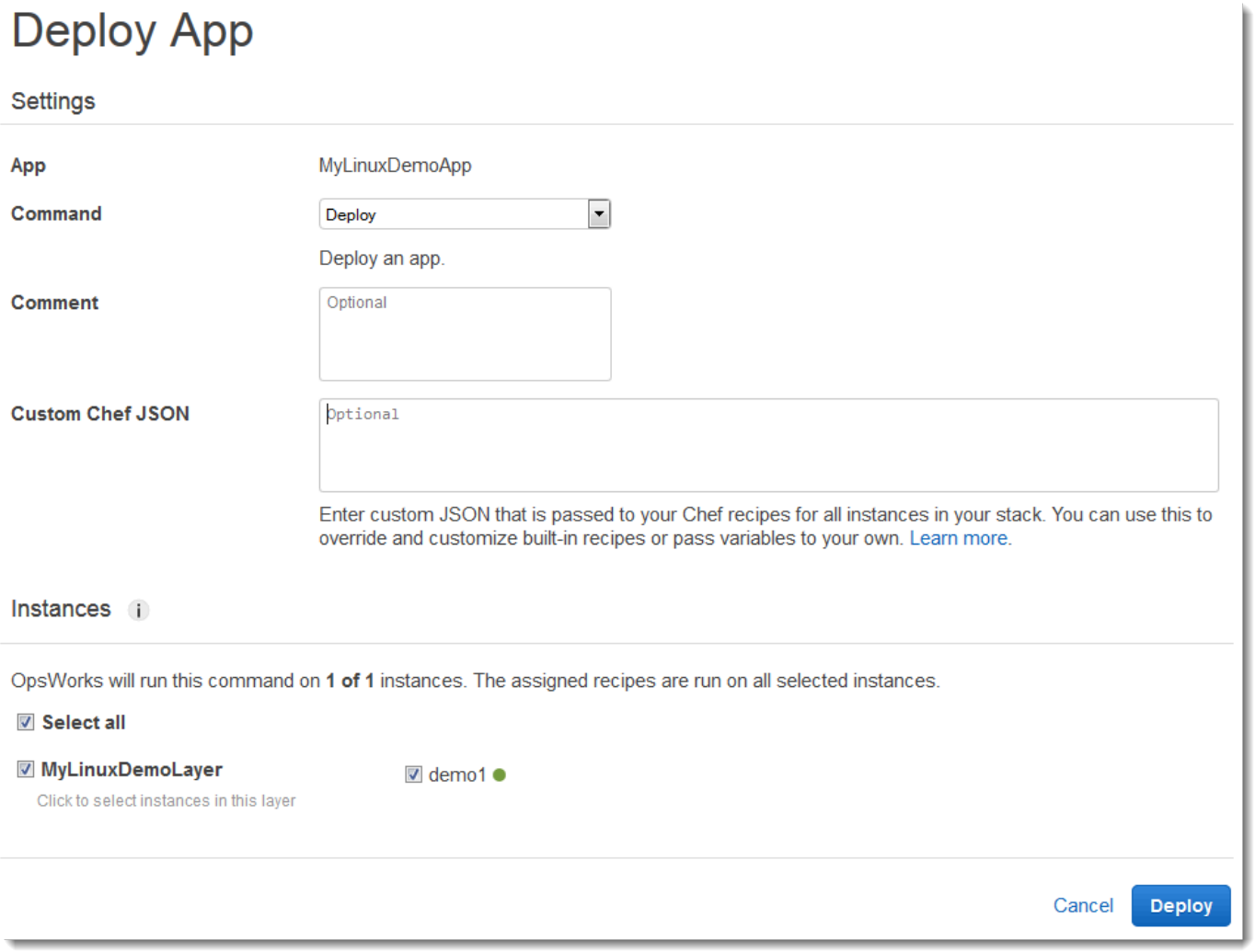
4. Choisissez Enregistrer. AWS OpsWorks Stacks ajoute la recette à l'événement du cycle de vie de déploiement de la couche.

Pour déployer l'application sur l'instance

1. Dans le panneau de navigation du service, sélectionnez Apps (Applications). La page Apps (Applications) s'affiche.
2. Pour MyLinuxDemoApp, pour Actions, choisissez déployer, comme indiqué dans la capture d'écran suivante :



3. Sur la page Deploy App (Déployer l'application), conservez les valeurs par défaut pour les éléments suivants :
   - Command (Commande) (Deploy)
   - Comment (Commentaire) (vide)
   - Settings (Paramètres), Advanced (Avancés), Custom Chef JSON (JSON Chef personnalisé) (vide)
   - Instances, avancées (cochée Tout sélectionner, cochée MyLinuxDemoLayer, démo1 cochée)
4. Vos résultats doivent correspondre à la capture d'écran suivante :



5. Choisissez Déployer. La page Déploiement MyLinuxDemoApp — déploiement s'affiche. Status (Statut) passe de running (en cours d'exécution) à successful (succès). Un cercle en rotation s'affiche à côté de demo1, qui se change en coche de couleur verte. Notez que ce processus peut prendre plusieurs minutes. Ne poursuivez pas tant que vous ne voyez pas Status (Statut) avec la valeur successful (succès), ainsi que la coche de couleur verte.
6. Vos résultats doivent correspondre à la capture d'écran suivante, sauf bien sûr pour Created at (Créé à), Completed at (Terminé à), Duration (Durée) et User (Utilisateur). Si Status (Statut) a la valeur failed (échec), pour résoudre les problèmes, pour Log (Journal), choisissez show (afficher) pour obtenir les détails relatifs à l'échec :



Vous avez maintenant déployé avec succès l'application sur l'instance.

Dans l'étape suivante, vous allez tester l'application déployée sur l'instance.

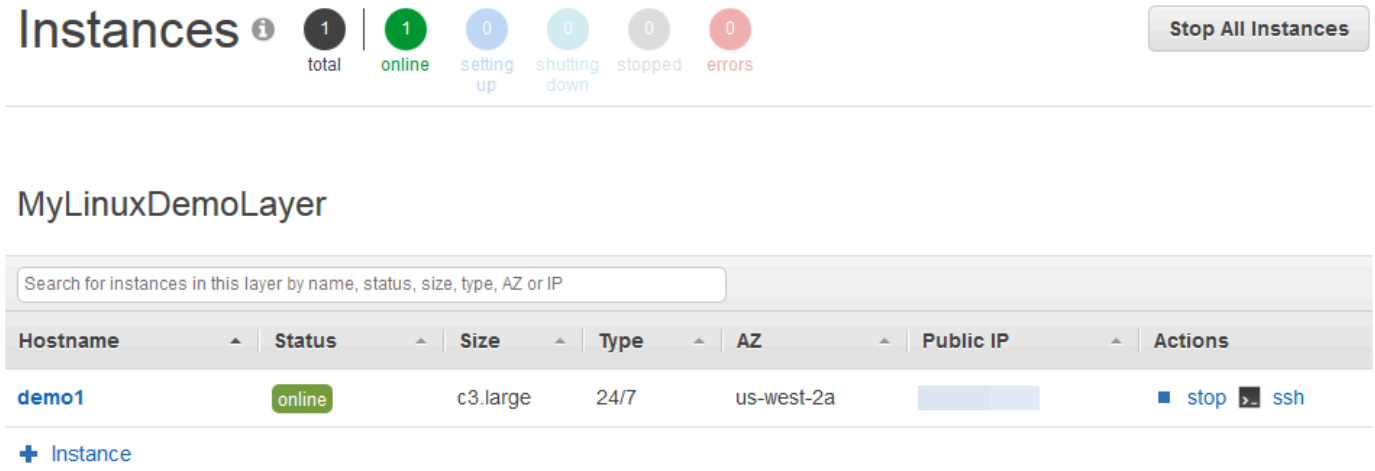
Étape 7 : Tester l'application déployée sur l'instance

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Maintenant, testez le déploiement de l'application sur l'instance.

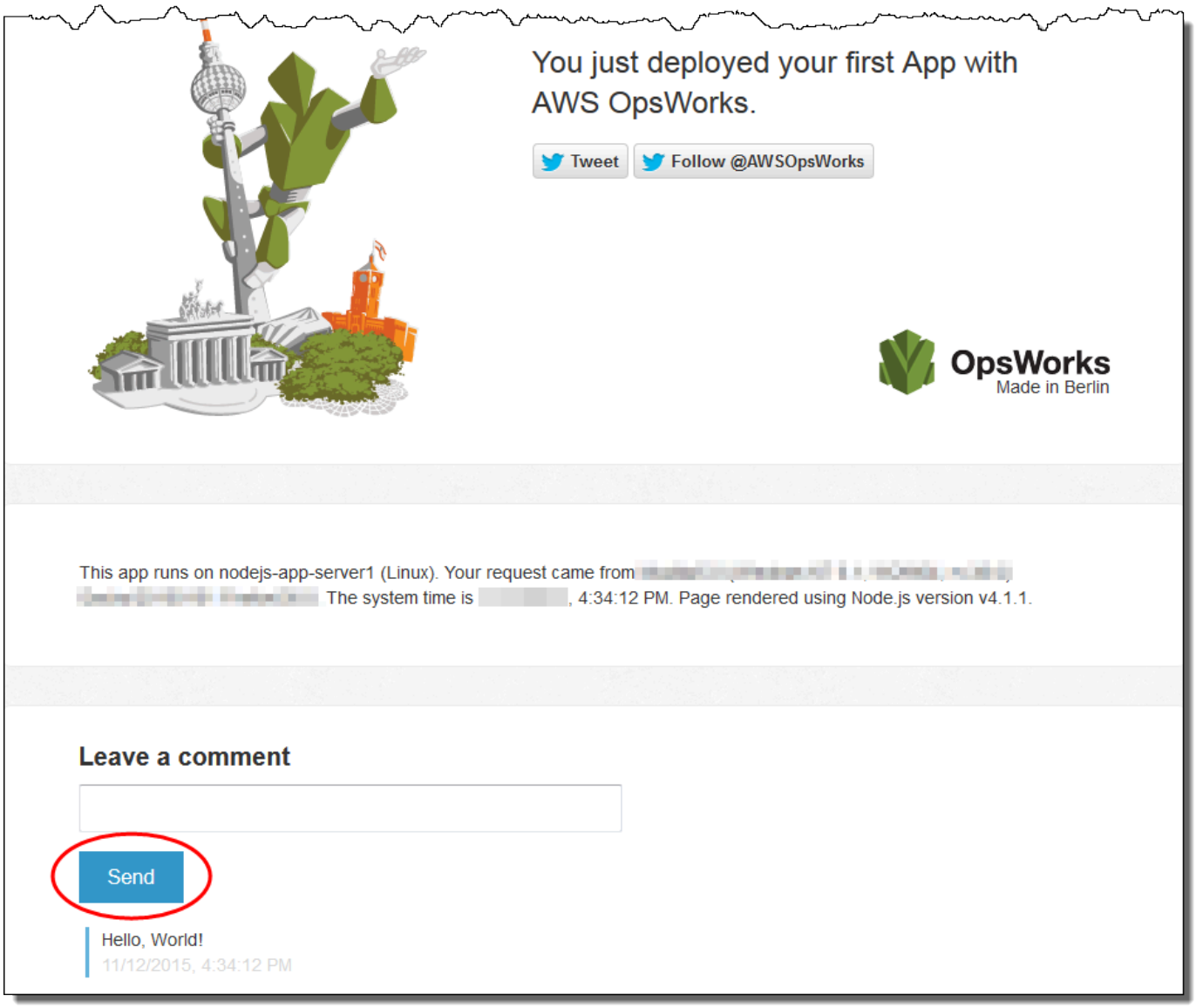
Pour tester le déploiement sur l'instance

1. Dans le panneau de navigation du service, choisissez Instances. La page Instances s'affiche.
2. Pour MyLinuxDemoLayer, pour demo1, pour Public IP, choisissez l'adresse IP :



Un nouvel onglet de navigateur web affiche l'application.

3. Sur la page web de félicitations, dans la zone de texte Leave a comment (Laisser un commentaire), entrez un commentaire, puis choisissez Send (Envoyer) pour tester l'application. L'application ajoute votre commentaire à la page web. Continuez à laisser des commentaires et à choisir Send (Envoyer) aussi souvent que vous le souhaitez :



4. Si vous avez un compte Twitter, choisissez Tweeter ou Suivre @ AWSOps Works, puis suivez les instructions qui s'affichent à l'écran pour tweeter à propos de l'application ou pour suivre @ AWSOps Works.

Vous avez testé avec succès l'application déployée sur l'instance.

À l'étape suivante, vous pouvez nettoyer les AWS ressources que vous avez utilisées pour cette procédure pas à pas. Cette étape suivante est facultative. Vous souhaiterez peut-être continuer à utiliser ces AWS ressources tout en continuant à en apprendre davantage sur AWS OpsWorks Stacks. Cependant, la conservation de ces AWS ressources peut entraîner des frais permanents sur

votre AWS compte. Si vous souhaitez conserver ces AWS ressources pour une utilisation ultérieure, vous avez maintenant terminé cette procédure pas à pas et vous pouvez passer àÉtapes suivantes.

Étape 8 : (Facultatif) Nettoyer

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Pour éviter d'encourir des frais supplémentaires sur votre AWS compte, vous pouvez supprimer les AWS ressources utilisées pour cette procédure pas à pas. Ces AWS ressources incluent la pile AWS OpsWorks Stacks et les composants de la pile. (Pour plus d'informations, consultez la section AWS OpsWorks Tarification.) Cependant, vous souhaiterez peut-être continuer à utiliser ces AWS ressources tout en continuant à en apprendre davantage sur AWS OpsWorks Stacks. Si vous souhaitez que ces AWS ressources restent disponibles, vous avez maintenant terminé cette procédure pas à pas et vous pouvez passer àÉtapes suivantes.

Le contenu stocké dans les ressources que vous avez créées pour cette procédure pas à pas peut contenir des informations d'identification personnelle. Si vous ne voulez plus que ces informations soient stockées par AWS, suivez les étapes décrites dans cette rubrique.

Pour supprimer l'application de la pile

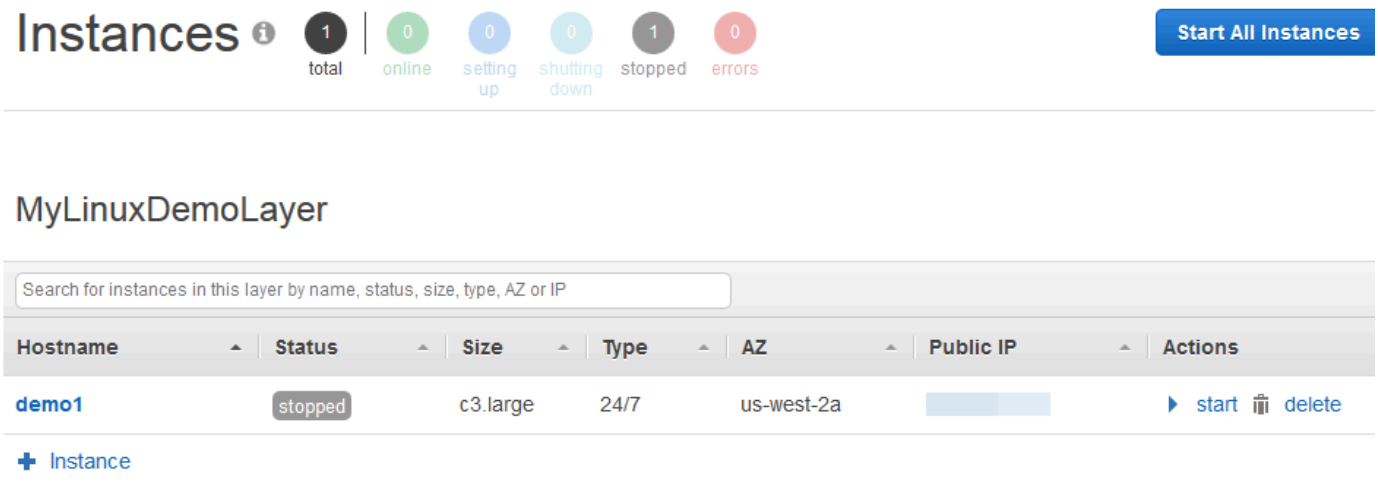
1. Dans la console AWS OpsWorks Stacks, dans le volet de navigation des services, choisissez Apps. La page Apps (Applications) s'affiche.
2. Pour MyLinuxDemoApp, pour Actions, choisissez Supprimer. Lorsque le message de confirmation s'affiche, choisissez Supprimer. AWS OpsWorks Stacks supprime l'application.

Pour supprimer l'instance de la pile

1. Dans le panneau de navigation du service, choisissez Instances. La page Instances s'affiche.
2. Pour MyLinuxDemoLayer, pour demo1, pour Actions, choisissez stop. Dans le message de confirmation, sélectionnez Stop. Les actions suivantes ont lieu.

- Status (Statut) passe de online (en ligne) à stopping (en cours d'arrêt) et, enfin, à stopped (arrêté).
- online (en ligne) passe de 1 à 0.
- shutting down (en cours de fermeture) passe de 0 à 1 et, enfin, à 0.
- stopped (arrêté) passe finalement de 0 à 1.

Ce processus peut prendre quelques minutes. Lorsque AWS OpsWorks Stacks est terminé, les résultats suivants sont affichés.



3. Pour Actions, choisissez delete (supprimer). Lorsque le message de confirmation s'affiche, choisissez Supprimer. AWS OpsWorks Stacks supprime l'instance et affiche le message No instances.

Pour supprimer la pile

1. Dans le panneau de navigation du service, choisissez Stack (Pile). La page MyLinuxDemoStack s'affiche.
2. Choisissez Supprimer pile. Lorsque le message de confirmation s'affiche, choisissez Supprimer. AWS OpsWorks Stacks supprime la pile et affiche la page du OpsWorks tableau de bord.

Vous pouvez éventuellement supprimer l'utilisateur et la paire de EC2 clés Amazon que vous avez utilisés pour cette procédure pas à pas, si vous ne souhaitez pas les réutiliser pour accéder à d'autres AWS services et EC2 instances. Pour obtenir des instructions, consultez Supprimer un utilisateur IAM, des paires de EC2 clés Amazon et des instances Linux.

Vous avez maintenant terminé cette procédure pas à pas. Pour de plus amples informations, veuillez consulter Étapes suivantes.

Étapes suivantes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Maintenant que vous avez terminé cette procédure pas à pas, vous pouvez en savoir plus sur l'utilisation de AWS OpsWorks Stacks :

- Explorez le livre de recettes et l'application que vous avez utilisés pour cette procédure pas à pas. Consultez En savoir plus : Explorer le livre de recettes utilisé dans cette procédure pas à pas et En savoir plus : Explorer l'application utilisée dans la procédure pas à pas.
- Entraînez-vous à utiliser AWS OpsWorks Stacks avec des instances Windows. Consultez Mise en route : Windows.
- Découvrez-en plus sur les piles en apprenant comment Créer une pile.
- Découvrez-en plus sur les couches en Modification de OpsWorks la configuration d'une couche.
- Découvrez-en plus sur les instances en Ajout d'une instance à une couche.
- Découvrez-en plus sur les applications en Déploiement d'applications.
- En savoir plus sur Livres de recettes et recettes.
- Créez vos propres livres de recettes. Consultez Mise en route : livres de recettes.
- Découvrez-en plus sur le contrôle de l'accès aux piles avec Sécurité et autorisations.

En savoir plus : Explorer le livre de recettes utilisé dans cette procédure pas à pas

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement

aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette rubrique décrit le livre de recettes utilisé par AWS OpsWorks Stacks pour la procédure pas à pas.

Un livre de recettes est un concept Chef. Les livres de recettes sont des fichiers d'archive qui contiennent les informations de configuration, telles que les recettes, les valeurs d'attribut, les fichiers, les modèles, les bibliothèques, les définitions et les ressources personnalisées. Une recette est également un concept Chef. Les recettes sont des instructions, écrites avec la syntaxe du langage Ruby, qui spécifient les ressources à utiliser et l'ordre dans lequel ces ressources sont appliquées. Pour plus d'informations, consultez À propos des livres de recettes et À propos des recettes sur le site web Découvrir Chef.

Pour voir le contenu du livre de recettes utilisé dans cette procédure pas à pas, extrayez le contenu du fichier opsworks-linux-demo-cookbooks-nodejs.tar.gz dans un répertoire vide de votre poste de travail local. (Vous pouvez également vous connecter à l'instance sur laquelle vous avez déployé le livre de recettes et explorer le contenu du répertoire `/var/chef/cookbooks`.)

Le fichier `default.rb` dans le répertoire `cookbooks/nodejs_demo/recipes` est l'emplacement où le livre de recettes exécute son code :

```
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
```

Voici ce que fait le fichier :

- `search(:aws_opsworks_app).first` utilise la recherche Chef pour rechercher des informations sur l'application qui sera déployée sur l'instance. Ces informations incluent des paramètres tels que le nom court de l'application et les détails du référentiel source. Comme une seule application a été déployée dans cette procédure pas à pas, la recherche Chef obtient ces paramètres depuis le premier élément d'information au sein de l'index de recherche `aws_opsworks_app` sur l'instance. Chaque fois qu'une instance est lancée, AWS OpsWorks Stacks stocke ces informations et d'autres informations connexes sous forme d'ensemble de sacs de données sur l'instance elle-même, et vous obtenez le contenu du sac de données via Chef search. Même si vous pouvez coder en dur ces paramètres dans la recette, l'utilisation des conteneurs de données et la recherche Chef constituent une approche plus solide. Pour plus d'informations sur les conteneurs de données, consultez AWS OpsWorks Référence du sac de données Stacks. Consultez aussi À propos des conteneurs de données sur le site web Découvrir Chef. Pour plus d'informations sur la recherche Chef, consultez À propos de la recherche sur le site web Découvrir Chef.
- La ressource `package` installe Git sur l'instance.
- La ressource `application` décrit et déploie les applications web :
  - `javascript`test la version du JavaScript moteur d'exécution à installer.
  - `environment` définit une variable d'environnement.
  - `git` obtient le code source du référentiel et de la branche spécifiés.
  - `app_path` est le chemin d'accès pour cloner le référentiel. Si le chemin n'existe pas sur l'instance, AWS OpsWorks Stacks le crée.
  - `link` crée un lien symbolique.
  - `npm_install` installe le Gestionnaire de package de nœud, le gestionnaire de package par défaut pour Node.js.
  - `npm_start` exécute Node.js.

Bien que AWS OpsWorks Stacks ait créé le livre de recettes utilisé pour cette procédure pas à pas, vous pouvez créer vos propres livres de recettes. Pour savoir comment procéder, veuillez consulter la section Mise en route : livres de recettes. De même, consultez À propos des livres de recettes, À propos des recettes et Découvrir les bases de Chef sur Ubuntu sur le site web Découvrir Chef et la section « Notre premier livre de recettes Chef » de Premières étapes avec Chef sur le site web Mise en route avec Chef.

En savoir plus : Explorer l'application utilisée dans la procédure pas à pas

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette rubrique décrit l'application que AWS OpsWorks Stacks déploie sur l'instance pour cette procédure pas à pas.

Pour voir le code source de l'application, extrayez le contenu du opsworks-windows-demo-nodejs GitHub référentiel dans un répertoire vide de votre poste de travail local. Vous pouvez également vous connecter à l'instance sur laquelle vous avez déployé le livre de recettes et explorer le contenu du répertoire `/srv/mylinuxdemoapp`.

Le fichier `index.js` contient le code le plus important pour l'application :

```
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);
});
```

Voici ce que fait le fichier :

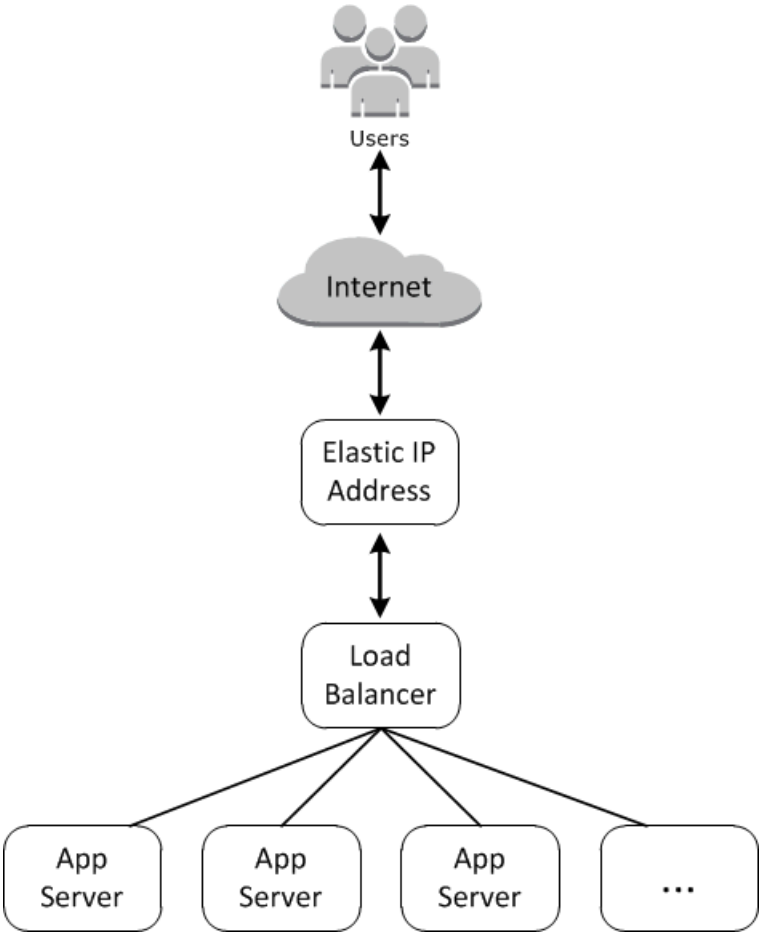
- `require` charge les modules qui contiennent un code dépendant que cette application web doit exécuter comme prévu.
- Les fonctions `add_comment` et `get_comments` écrivent des informations sur le fichier `comments.json` et en lisent à partir de ce fichier.
- Pour plus d'informations sur `app.get`, `app.listen`, `app.post`, `app.set`, et `app.use`, consultez Référence express des API.

Pour apprendre à créer et à empaqueter votre application pour le déploiement, consultez Source de l'application.

Mise en route sur les piles Windows

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les applications basées sur le cloud nécessitent généralement un groupe de ressources connexes (serveurs d'applications, serveurs de base de données, etc.) qui doivent être créées et gérées collectivement. Cet ensemble d'instances est appelé une pile. Une simple pile d'application peut se présenter comme suit.



L'architecture de base comprend les éléments suivants :

- Une adresse IP Elastic pour recevoir les demandes utilisateur.
- Un équilibreur de charge pour répartir les demandes entrantes de manière égale entre les serveurs d'applications.
- Un ensemble d'instances serveur d'applications, en nombre suffisant pour gérer le trafic.

En outre, vous avez généralement besoin d'une solution pour répartir les applications sur les serveurs d'applications, gérer les autorisations utilisateur, etc.

AWS OpsWorks Stacks fournit un moyen simple et direct de créer et de gérer des piles ainsi que les applications et ressources associées. Ce chapitre présente les principes de base de AWS OpsWorks Stacks, ainsi que certaines de ses fonctionnalités les plus sophistiquées, en vous expliquant le processus de création de la pile de serveurs d'applications dans le diagramme. Il utilise un modèle de développement incrémentiel que AWS OpsWorks Stacks rend facile à suivre : configurez une pile

de base et, une fois qu'elle fonctionne correctement, ajoutez des composants jusqu'à ce que vous obteniez une implémentation complète.

- Étape 1 : Exécuter les opérations prérequises montre comment se préparer à démarrer la procédure pas à pas.
- Étape 2 : Créer une pile serveur d'applications de basemontre comment créer une pile de base pour prendre en charge les services Internet (IIS) et déployer une application sur le serveur.
- Étape 3 : Extensification IISExample montre comment agrandir une pile pour gérer la charge accrue en ajoutant d'autres serveurs d'applications, un équilibreur de charge pour répartir le trafic entrant et une adresse IP Elastic pour recevoir des demandes entrantes.

Rubriques

- Étape 1 : Exécuter les opérations prérequises
- Étape 2 : Créer une pile serveur d'applications de base
- Étape 3 : Extensification IISExample
- Étapes suivantes

Étape 1 : Exécuter les opérations prérequises

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Complétez les étapes de configuration suivantes avant de commencer la procédure pas à pas. Ces étapes de configuration incluent la création d'un AWS compte, la création d'un utilisateur administratif et l'attribution d'autorisations d'accès à AWS OpsWorks Stacks.

Si vous avez déjà terminé les procédures pas à pas Mise en route : exemple ou Mise en route : Linux, vous remplissez les conditions requises pour cette procédure pas à pas et pouvez passer directement à Étape 2 : Créer une pile serveur d'applications de base.

Rubriques

- Inscrivez-vous pour un Compte AWS
- Création d'un utilisateur doté d'un accès administratif
- Attribuer des autorisations d'accès au service
- Assurez-vous que les utilisateurs de AWS OpsWorks Stacks sont ajoutés à votre domaine

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

   Une partie de la procédure d'inscription consiste à recevoir un appel téléphonique ou un message texte et à saisir un code de vérification sur le clavier du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à https://aws.amazon.com/et en choisissant Mon compte.

Création d'un utilisateur doté d'un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez Utilisateur racine d'un compte AWS AWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez Connexion en tant qu'utilisateur racine dans le Guide de l'utilisateur Connexion à AWS .
2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, voir Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d' AWS IAM Identity Center dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu'utilisateur doté d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

  Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribution d'un accès à d'autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d'autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

Pour obtenir des instructions, consultez Création d'un ensemble d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

   Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l'utilisateur AWS IAM Identity Center .

Attribuer des autorisations d'accès au service

Activez l'accès au service AWS OpsWorks Stacks (et aux services connexes sur lesquels AWS OpsWorks Stacks s'appuie) en ajoutant les `AmazonS3FullAccess` autorisations `AWSOpsWorks_FullAccess` et à votre rôle ou à votre utilisateur.

Pour plus d'informations sur l'ajout d'autorisations, consultez la section Ajout d'autorisations d'identité IAM (console).

Assurez-vous que les utilisateurs de AWS OpsWorks Stacks sont ajoutés à votre domaine

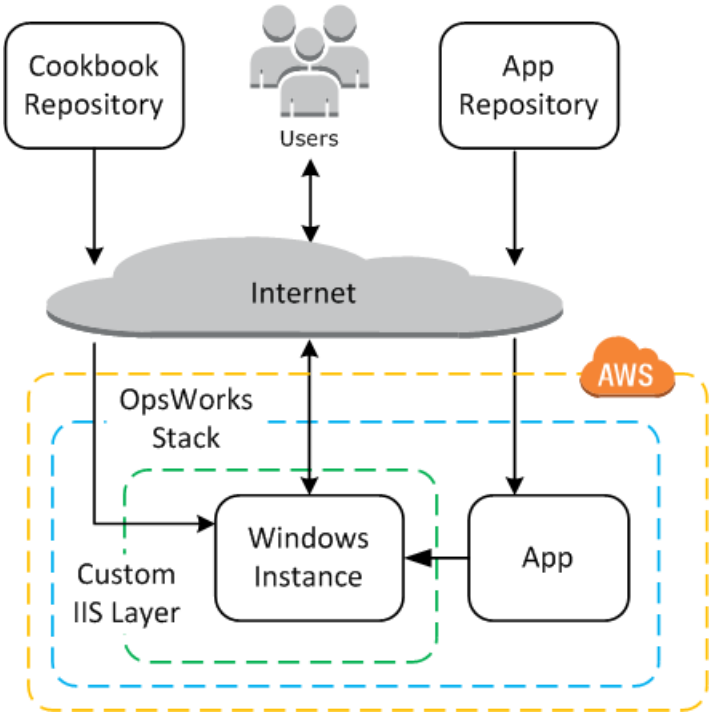
Dans une pile Chef 12.2, le livre de recettes `aws_opsworks_users` inclus crée des utilisateurs qui ont un accès SSH et RDP (Remote Desktop Protocol) aux instances Windows. Lorsque vous associez des instances Windows de votre pile à un domaine Active Directory, l'exécution de ce livre de recettes peut échouer si les utilisateurs de AWS OpsWorks Stacks n'existent pas dans Active Directory. Si les utilisateurs ne sont pas reconnus dans Active Directory, les instances peuvent passer à un état `setup failed` lorsque vous les redémarrez après leur jonction à un domaine. Pour les instances Windows jointes à un domaine, il ne suffit pas d'accorder aux utilisateurs de AWS OpsWorks Stacks l' SSH/RDP accès sur la page des autorisations utilisateur.

Avant de joindre des instances Windows d'une pile Chef 12.2 à un domaine Active Directory, assurez-vous que tous les utilisateurs AWS OpsWorks Stacks de la pile Windows sont membres du domaine. La meilleure méthode consiste à configurer l'identité fédérée avec IAM avant de créer votre stack Windows, puis à importer des utilisateurs fédérés dans AWS OpsWorks Stacks avant de joindre les instances de votre stack à un domaine. Pour plus d'informations sur la procédure à suivre, consultez Activer la fédération sur AWS à l'aide de Windows Active Directory, ADFS et SAML 2.0 dans le blog de sécurité AWS, et Fédérer les utilisateurs existants dans le guide de l'utilisateur IAM.

Étape 2 : Créer une pile serveur d'applications de base

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une pile serveur d'applications de base se compose d'une seule instance serveur d'applications avec une adresse IP publique pour recevoir les demandes utilisateur. Le code de l'application et les fichiers associés sont stockés dans un référentiel distinct et déployés à partir de là vers le serveur. Le schéma suivant illustre une telle pile.



La pile comprend les éléments suivants :

- Une couche, qui représente un groupe d'instances et spécifie comment elles doivent être configurées.

  La couche de cet exemple représente un groupe d'instances IIS.
- Une instance, qui représente une EC2 instance Amazon.

Dans ce cas, la couche configure une seule instance pour exécuter IIS, mais les couches peuvent avoir n'importe quel nombre d'instances.

- Une application, qui contient les informations requises pour installer une application sur l'instance.
- Un livre de recettes, qui contient les recettes Chef personnalisées qui prennent en charge la couche IIS personnalisée. Le livre de recettes et le code de l'application sont stockés dans des référentiels distants, tels qu'un fichier d'archive dans un compartiment Amazon S3 ou un référentiel Git.

Les sections suivantes décrivent comment utiliser la console AWS OpsWorks Stacks pour créer la pile et déployer l'application.

Rubriques

- Étape 2.1 : Créer la pile
- Étape 2.2 : Autoriser l'accès RDP
- Étape 2.3 : Implémenter un livre de recettes personnalisé
- Étape 2.4 : Ajouter une couche IIS
- Étape 2.5 : Déployer une application

Étape 2.1 : Créer la pile

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous démarrez un projet AWS OpsWorks Stacks en créant une pile, qui fait office de conteneur pour vos instances et autres ressources. La configuration de la pile spécifie certains paramètres de base, telles que la région AWS et le système d'exploitation par défaut, qui sont partagés par toutes les instances de la pile.

Pour créer une pile

1. Ajouter une pile

   Si vous ne l'avez pas déjà fait, connectez-vous à la console AWS OpsWorks Stacks.

   - Si le compte ne possède aucune pile existante, vous verrez la OpsWorks page Bienvenue sur AWS ; choisissez Ajouter votre première pile.
   - Sinon, vous verrez le tableau de bord AWS OpsWorks Stacks, qui répertorie les stacks de votre compte ; choisissez Add Stack.

2. Configurer la pile

   Sur la page Add Stack (Ajouter une pile), choisissez Chef 12 stack (Pile Chef 12) et spécifiez les paramètres suivants :

   Nom de la pile

   Entrez un nom pour votre pile, qui peut contenir des caractères alphanumériques (a—z, A—Z et 0—9) et des tirets (-). L'exemple de pile de cette procédure est nommé **`IISWalkthrough`**.

   Région

   Sélectionnez USA West (Oregon) comme région de la pile.

   Vous pouvez créer une pile dans n'importe quelle région, mais nous recommandons US West (Oregon) pour les didacticiels.

   Système d'exploitation par défaut

   Choisissez Windows, puis spécifiez Microsoft Windows Server 2022 Base, qui est le paramètre par défaut.

   Utiliser les livres de recettes Chef personnalisés

   Dans le cadre de cette procédure, spécifiez No (Non) pour cette option.

3. Choisissez Advanced (Avancé) pour confirmer que vous avez un rôle IAM et le profil d'instance IAM par défaut sélectionnés.

   Rôle IAM

   Spécifiez le rôle IAM (AWS Identity and Access Management) de la pile. AWS OpsWorks Stacks doit accéder à d'autres services AWS pour effectuer des tâches telles que la création

et la gestion d' EC2 instances Amazon. Le rôle IAM indique le rôle que AWS OpsWorks Stacks assume pour travailler avec d'autres services AWS en votre nom. Pour de plus amples informations, veuillez consulter Permettre à AWS OpsWorks Stacks d'agir en votre nom.

- Si votre compte possède déjà un rôle AWS OpsWorks Stacks IAM, vous pouvez le sélectionner dans la liste.

  Si le rôle a été créé par AWS OpsWorks Stacks, il sera nommé`aws-opsworks-service-role`.

- Sinon, sélectionnez Nouveau rôle IAM pour demander à AWS OpsWorks Stacks de créer un nouveau rôle pour vous avec les autorisations appropriées.

  Remarque : si vous avez les autorisations d'accès complet AWS OpsWorks Stacks, la création d'un nouveau rôle nécessite plusieurs autorisations IAM supplémentaires. Pour de plus amples informations, veuillez consulter Exemples de stratégies.

4. Acceptez les valeurs par défaut pour les autres paramètres, puis choisissez Add Stack (Ajouter une pile). Pour plus d'informations sur les différents paramètres de pile, consultez Créer une pile.

Étape 2.2 : Autoriser l'accès RDP

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Maintenant que vous avez créé une pile, vous allez créer une couche et ajouter une instance Windows à la couche. Cependant, avant de pouvoir le faire, vous devez configurer la pile afin de pouvoir utiliser RDP pour vous connecter aux instances de la couche personnalisée. Pour ce faire, vous devez effectuer les opérations suivantes :

- Ajouter une règle de trafic entrant au groupe de sécurité qui contrôle l'accès RDP.
- Définissez vos autorisations AWS OpsWorks Stacks pour cette pile afin de permettre l'accès RDP.

Lorsque vous créez la première pile dans une région, AWS OpsWorks Stacks crée un ensemble de groupes de sécurité. Ils en incluent un nommé quelque chose comme`AWS-OpsWorks-RDP-Server`, que AWS OpsWorks Stacks attache à toutes les instances Windows pour permettre l'accès RDP. Cependant, comme, par défaut, ce groupe de sécurité n'a pas de règles, vous devez ajouter une règle entrante pour autoriser l'accès RDP à vos instances.

Pour autoriser l'accès RDP

1. Ouvrez la EC2 console Amazon, définissez-la sur la région de la pile et choisissez Security Groups dans le volet de navigation.
2. Choisissez AWS- OpsWorks -RDP-Server, choisissez l'onglet Inbound, puis sélectionnez Modifier.
3. Choisissez Add Rule (Ajouter une règle) et spécifiez les paramètres suivants :

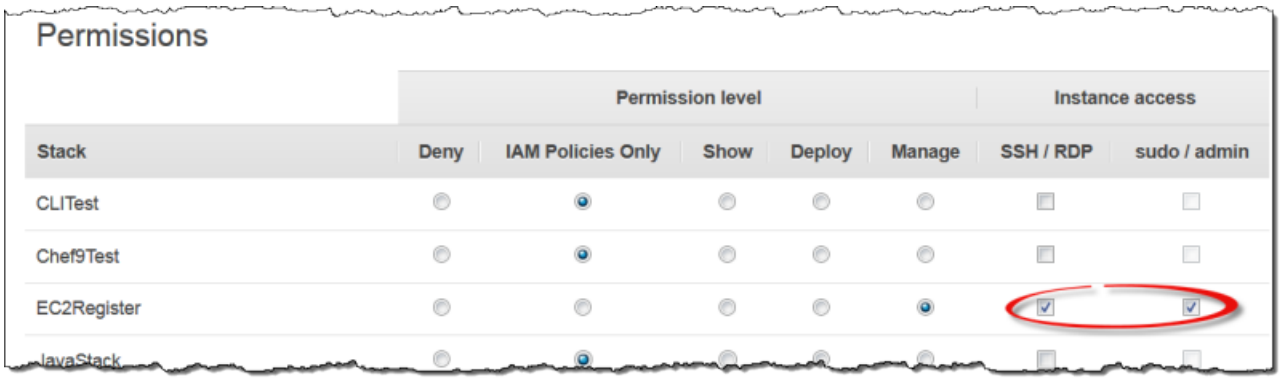
   - Tapez : RDP.
   - Source — Les adresses IP sources autorisées.

     Généralement, vous autorisez les demandes RDP entrantes de votre adresse IP ou plages d'adresses IP spécifiée (habituellement, votre plage d'adresses IP d'entreprise). A des fins d'apprentissage, il suffit souvent de spécifier 0.0.0.0/0, qui autorise l'accès RDP à partir de n'importe quelle adresse IP.

Le groupe de sécurité autorise l'instance à recevoir les demandes de connexion RDP, mais ce n'est que la moitié de l'histoire. Un utilisateur ordinaire se connectera à l'instance à l'aide d'un mot de passe fourni par AWS OpsWorks Stacks. Pour que AWS OpsWorks Stacks génère ce mot de passe, vous devez autoriser explicitement l'accès RDP à l'utilisateur.

Pour autoriser l'accès RDP pour un utilisateur

1. Dans le tableau de bord AWS OpsWorks Stacks, choisissez la IISWalkthroughpile.
2. Dans le panneau de navigation de la pile, sélectionnez Permissions (Autorisations).
3. Sur la page des autorisations, choisissez Edit (Modifier).
4. Dans la liste des utilisateurs, cochez la case SSH/RDP pour l'utilisateur auquel vous souhaitez accorder les autorisations nécessaires. Si vous voulez que l'utilisateur dispose également des autorisations d'administrateur, sélectionnez sudo/admin.



5. Choisissez Enregistrer.

L'utilisateur peut ensuite obtenir un mot de passe et l'utiliser pour se connecter à l'instance, comme décrit ultérieurement.

> **Note**
>
> Vous aussi pouvez vous connecter en tant qu'administrateur. Pour de plus amples informations, veuillez consulter Connexion en tant qu'Administrateur.

Étape 2.3 : Implémenter un livre de recettes personnalisé

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Même si une pile est essentiellement un conteneur d'instances, vous n'ajoutez pas directement les instances à une pile. Vous ajoutez une ou plusieurs couches, chacune d'elles représentant un groupe d'instances connexes, puis ajoutez les instances aux couches.

Une couche est essentiellement un plan que AWS OpsWorks Stacks utilise pour créer un ensemble d' EC2instances Amazon avec la même configuration. Une instance commence par une version de

base du système d'exploitation, puis la couche de l'instance effectue une grande variété de tâches sur l'instance afin d'implémenter cette empreinte :

- Création de répertoires et de fichiers
- Gestion des utilisateurs
- Installation et configuration de logiciels
- Arrêt ou démarrage de serveurs
- Déploiement du code d'application et des fichiers associés.

Une couche exécute des tâches sur des instances en exécutant des recettes Chef, c'est-à-dire des recettes en abrégé. Une recette est une application Ruby qui utilise le langage spécifique à un domaine (DSL) de Chef pour décrire l'état final de l'instance. Avec AWS OpsWorks Stacks, chaque recette est généralement affectée à l'un des événements du cycle de vie de la couche : installation, configuration, déploiement, dédéploiement et arrêt. Lorsqu'un événement du cycle de vie se produit sur une instance, AWS OpsWorks Stacks exécute les recettes de l'événement pour effectuer les tâches appropriées. Par exemple, l'événement Setup se produit une fois le démarrage d'une instance terminé. AWS OpsWorks Stacks exécute ensuite les recettes d'installation, qui exécutent généralement des tâches telles que l'installation et la configuration du logiciel serveur et le démarrage des services associés.

AWS OpsWorks Stacks fournit à chaque couche un ensemble de recettes intégrées qui exécutent des tâches standard. Vous pouvez étendre les fonctionnalités d'une couche en implémentant les recettes personnalisées pour exécuter des tâches supplémentaires et en les attribuant aux événements de cycle de vie de la couche. Les piles Windows prennent en charge les couches personnalisées, qui ont un ensemble minimal de recettes n'effectuant que quelques tâches de base. Pour ajouter des fonctionnalités à vos instances Windows, vous devez implémenter des recettes personnalisées pour pouvoir installer un logiciel, déployer des applications, etc. Cette rubrique décrit comment créer une simple couche personnalisée pour prendre en charge les instances IIS.

Rubriques

- Rapide présentation des livres de recettes et des recettes
- Implémenter une recette pour installer et démarrer IIS
- Activer le livre de recettes personnalisé

Rapide présentation des livres de recettes et des recettes

Une recette définit un ou plusieurs aspects de l'état attendu d'une instance : les répertoires qu'elle doit avoir, les packages logiciels qui doivent être installés, les applications qui doivent être déployées, etc. Les recettes sont regroupées dans un livre de recettes, qui contient généralement une ou plus recettes connexes, plus les fichiers associés tels que les modèles pour la création des fichiers de configuration.

Ce sujet est une introduction très élémentaire aux recettes, juste destinée à vous montrer comment implémenter un livre de recettes pour prendre en charge une couche IIS personnalisée simple. Pour une introduction plus générale aux livres de recettes, consultez Livres de recettes et recettes. Pour une introduction détaillée du didacticiel à l'implémentation des livres de recettes, y compris certaines rubriques spécifiques à Windows, consultez Les bases des livres de recettes.

Les recettes Chef sont techniquement des applications Ruby, mais la plus grande partie du code, si ce n'est la totalité, est écrit en DSL Chef. Le DSL se compose pour l'essentiel d'un ensemble de ressources, qui vous permettent de spécifier par déclaration un aspect de l'état des instances. Par exemple, une `directory` ressource définit un répertoire à ajouter au système. L'exemple suivant définit un répertoire `C:\data` avec les droits de contrôle total qui appartient à l'utilisateur spécifié et n'hérite pas les droits du répertoire parent.

```
directory 'C:\data' do
  rights :full_control, 'WORKGROUP\username'
  inherits false
  action :create
end
```

Lorsque Chef exécute une recette, il exécute chaque ressource en transmettant les données à un fournisseur associé, objet Ruby qui gère les détails de la modification de l'état de l'instance. Dans ce cas, le fournisseur crée un nouveau répertoire avec la configuration spécifiée.

Le livre de recettes personnalisé de la couche IIS personnalisée doit effectuer les tâches suivantes :

- Installer la fonctionnalité IIS et démarrer le service.

  Généralement, vous effectuez cette tâche lors de l'installation, juste après que l'instance a fini de démarrer.
- Déployer une application sur l'instance, une simple page HTML dans cet exemple.

Généralement, vous effectuez cette tâche lors de l'installation. Cependant, comme, en règle générale, les applications doivent être mises à jour régulièrement, vous devez également déployer les mises à jour pendant que l'instance est en ligne.

Vous pouvez n'avoir qu'une seule recette pour effectuer toutes ces tâches. Cependant, l'approche privilégiée consiste à avoir des recettes distinctes pour les tâches d'installation et de déploiement. De cette façon, vous pouvez déployer les mises à jour de l'application à tout moment sans exécuter aussi le code d'installation. La procédure suivante explique comment configurer un livre de recettes pour prendre en charge une couche IIS personnalisée. Les rubriques suivantes montrent comment implémenter les recettes.

Mise en route

1. Créez un répertoire nommé `iis-cookbook` sur un emplacement approprié de votre ordinateur.
2. Ajoutez un fichier `metadata.rb` avec le contenu suivant à `iis-cookbook`.

   ```
   name "iis-cookbook"
   version "0.1.0"
   ```

   Cet exemple utilise un fichier `metadata.rb` minimal. Pour plus d'informations sur l'utilisation de ce fichier, consultez metadata.rb.
3. Ajoutez un répertoire `recipes` à `iis-cookbook`.

   Ce répertoire, qui doit s'appeler `recipes`, contient les recettes du livre de recettes.

En général, les livres de recettes peuvent contenir une grande variété d'autres répertoires. Par exemple, si une recette utilise un modèle pour créer un fichier de configuration, le modèle se trouve généralement dans le répertoire `templates\default`. Comme le livre de recettes de cet exemple se compose entièrement de recettes, il n'a besoin d'aucun autre répertoire. En outre, cet exemple utilise un seul livre de recettes, mais vous pouvez en utiliser autant que nécessaire ; plusieurs livres de recettes sont souvent préférables pour les projets complexes. Par exemple, vous pouvez avoir des livres de recettes distincts pour les tâches d'installation et de déploiement. Pour obtenir plus d'exemples de livres de recettes, consultez Livres de recettes et recettes.

Implémenter une recette pour installer et démarrer IIS

IIS est une fonctionnalité Windows, un ensemble de composants système facultatifs que vous pouvez installer sur Windows Server. Vous pouvez faire en sorte qu'une recette installe IIS à l'aide de l'une des solutions suivantes :

- En utilisant une ressource `powershell_script` pour exécuter l'applet de commande `Install-WindowsFeature`.
- En utilisant la ressource Chef du livre de recettes de Windows `windows_feature`.

  Le livre de recettes de `windows` contient un ensemble de ressources dont les fournisseurs utilisent Gestion et maintenance des images de déploiement (DISM) pour effectuer une grande variété de tâches sur les instances Windows, y compris l'installation de fonctionnalités.

> **Note**
>
> `powershell_script` figure parmi les ressources les plus utiles des recettes Windows. Vous pouvez l'utiliser pour effectuer diverses tâches sur une instance en exécutant un PowerShell script ou une applet de commande. Il est particulièrement utile pour les tâches qui ne sont pas prises en charge par une ressource Chef.

Cet exemple exécute un PowerShell script pour installer et démarrer le serveur Web (IIS). Le livre de recettes `windows` est décrit plus loin. Pour obtenir un exemple d'utilisation de `windows_feature` pour installer IIS, consultez Installation d'une fonctionnalité de Windows : IIS.

Ajoutez une recette nommée `install.rb` avec le contenu suivant au répertoire `recipes` du live de recettes.

```
powershell_script 'Install IIS' do
  code 'Install-WindowsFeature Web-Server'
  not_if "(Get-WindowsFeature -Name Web-Server).Installed"
end

service 'w3svc' do
  action [:start, :enable]
end
```

La recette contient deux ressources.

powershell_script

`powershell_script`exécute le PowerShell script ou l'applet de commande spécifié. L'exemple a les paramètres d'attribut suivants :

- `code`— Les PowerShell applets de commande à exécuter.

  Cet exemple exécute une applet de commande `Install-WindowsFeature`, laquelle installe le serveur Web (IIS). En général, comme l'attribut `code` peut avoir un nombre quelconque de lignes, vous pouvez exécuter autant d'applets de commande que nécessaire.

- `not-if`— Un attribut de protection qui garantit que la recette installe IIS uniquement s'il n'a pas encore été installé.

  Comme vous voulez généralement que les recettes soient idempotentes, elles ne perdent pas de temps à exécuter la même tâche plusieurs fois.

Chaque ressource possède une action, qui spécifie l'action que le fournisseur doit prendre. Il n'y a aucune action explicite pour cet exemple. Le fournisseur exécute donc l'`:run`action par défaut, qui exécute le PowerShell script spécifié. Pour de plus amples informations, veuillez consulter Exécution d'un PowerShell script Windows.

web

Un `service` gère un service, le service serveur Web IIS (W3SVC) dans le cas présent. L'exemple utilise les attributs par défaut et spécifie deux actions, `:start` et `:enable`, qui démarrent et activent IIS.

> **Note**
>
> Si vous souhaitez installer un logiciel qui utilise un programme d'installation de packages, tel que MSI, vous pouvez utiliser une ressource `windows_package`. Pour de plus amples informations, veuillez consulter Installation d'un package.

Activer le livre de recettes personnalisé

AWS OpsWorks Stacks exécute des recettes à partir d'un cache local sur chaque instance. Pour exécuter vos recettes personnalisées, vous devez effectuer les opérations suivantes :

- Archivez le livre de recettes dans un référentiel distant.

AWS OpsWorks Stacks télécharge les livres de recettes de ce référentiel vers le cache local de chaque instance.

- Modifier la pile pour activer les livres de recettes personnalisés.

  Comme les livres de recettes personnalisés sont désactivés par défaut, vous devez activer les livres de recettes personnalisés pour la pile, ainsi que fournir l'URL du référentiel et les informations associées.

AWS OpsWorks Stacks prend en charge les archives S3 et les référentiels Git pour les livres de recettes personnalisés ; cet exemple utilise une archive S3. Pour de plus amples informations, veuillez consulter Référentiels de livres de recettes.

Pour utiliser une archive S3

1. Créez une archive `.zip` du répertoire `iis-cookbook`.

   AWS OpsWorks Stacks prend également en charge les archives `.tgz` (tar compressé au format gzip) pour les piles Windows.

2. Téléchargez l'archive dans un compartiment S3 dans la région de l'ouest des États-Unis (Californie du Nord) et rendez le fichier public. Vous pouvez également utiliser des archives privées S3, mais les archives publiques sont suffisantes pour cet exemple et un peu plus simples à utiliser.

   a. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

   b. Si vous n'avez pas encore de compartiment`us-west-1`, choisissez Create Bucket et créez un bucket dans la région USA Ouest (Californie du Nord).

   c. Dans la liste des compartiments, choisissez le nom du compartiment sur lequel vous souhaitez charger le fichier, puis choisissez Upload (Charger).

   d. Choisissez Add Files (Ajouter des fichiers).

   e. Sélectionnez le fichier d'archive à charger, puis choisissez Open (Ouvrir).

   f. En bas de la boîte de dialogue Upload - Select Files and Folders (Charger - Sélectionner les fichiers et dossiers), choisissez Set Details (Définir les détails).

   g. En bas de la boîte de dialogue Set Details (Définir les détails), choisissez Set Permissions (Définir les autorisations).

h. Dans la boîte de dialogue Set Permissions (Définir les autorisations), choisissez Make everything public (Rendre tout public).

i. En bas de la boîte de dialogue Set Permissions (Définir les autorisations), choisissez Start Upload (Commencer le chargement). Lorsque le chargement est terminé, le fichier `iis-cookbook.zip` s'affiche dans votre compartiment.

j. Choisissez le compartiment, puis choisissez l'onglet Properties (Propriétés) du compartiment. En regard de Link (Lien), enregistrez l'URL du fichier d'archives en vue d'une utilisation ultérieure.

Pour plus d'informations sur le chargement de fichiers dans un compartiment Amazon S3, consultez Comment télécharger des fichiers et des dossiers dans un compartiment S3 ? dans le guide de l'utilisateur de la console Amazon S3.

> **Important**
>
> Jusque-là, la procédure pas à pas ne vous a pris que peu de temps ; le service AWS OpsWorks Stacks lui-même est gratuit. Cependant, vous devez payer pour toutes les ressources AWS que vous utilisez, telles que le stockage Amazon S3. Dès que vous chargez l'archive, vous commencez à être facturé. Pour plus d'informations, consultez Tarification AWS.

Pour activer les livres de recettes personnalisés pour la pile

1. Dans la console AWS OpsWorks Stacks, choisissez Stack dans le volet de navigation, puis sélectionnez Stack Settings en haut à droite.
2. Dans le coin supérieur droit de la page Settings (Paramètres), choisissez Edit (Modifier).
3. Sur la page Settings (Paramètres), définissez Use custom Chef cookbooks (Utiliser les livres de recettes Chef personnalisés) avec la valeur Yes (Oui) et entrez les informations suivantes :
   - Type de référentiel : archive S3.
   - URL du référentiel : URL S3 du fichier d'archive du livre de recettes que vous avez enregistré précédemment.
4. Choisissez Save (Enregistrer) pour mettre à jour la configuration de la pile.

AWS OpsWorks Stacks installe votre livre de recettes personnalisé sur toutes les nouvelles instances. Notez qu' AWS OpsWorks Stacks n'installe pas ou ne met pas à jour automatiquement les livres de recettes personnalisés sur les instances en ligne. Vous pouvez le faire manuellement, comme indiqué plus tard.

Étape 2.4 : Ajouter une couche IIS

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Votre livre de recettes a une recette qui installe et démarre simplement IIS. Cela suffit pour créer la couche et vérifier que vous avez une instance IIS qui fonctionne. Plus tard, vous ajouterez la fonctionnalité de déploiement d'application à la couche.

Créer une couche

Vous commencez par ajouter une couche à la pile. Ensuite, vous ajoutez les fonctionnalités à cette couche en affectant des recettes personnalisées aux événements de cycle de vie appropriés.

Pour ajouter une couche IIS à la pile

1. Choisissez Layers (Couches) dans le panneau de navigation, puis Add a layer (Ajouter une couche).
2. Configurez la couche comme suit :

   - Nom — **`IISExample`**
   - Nom abrégé — **`iisexample`**

     AWS OpsWorks Stacks utilise le nom abrégé pour identifier la couche en interne. Vous utilisez aussi le nom court pour identifier la couche dans les recettes, même si cet exemple ne le fait pas. Vous pouvez spécifier n'importe quel nom court, mais il ne peut se composer que de caractères alphanumériques en lettres minuscules et d'un petit nombre de signes de ponctuation. Pour de plus amples informations, veuillez consulter Couches personnalisées.

3. Choisissez Add Layer (Ajouter une couche).

Si vous deviez ajouter une instance à ce IISWalkthrough stade et la démarrer, AWS OpsWorks Stacks installerait automatiquement les livres de recettes mais elle ne s'exécuterait pas. `install.rb` Une fois qu'une instance est en ligne, vous pouvez exécuter les recettes manuellement grâce à la commande de pile Execute Recipes. Cependant, une meilleure approche consiste à attribuer la recette à l'un des événements du cycle de vie de la couche. AWS OpsWorks Stacks exécute ensuite automatiquement la recette au moment approprié du cycle de vie de l'instance.

Installez et démarrez IIS dès que le démarrage de l'instance est terminé. Pour ce faire, attribuez `install.rb` à l'événement `Setup` de la couche.

Pour attribuer la recette à un événement du cycle de vie

1. Choisissez Layers (Couches) dans le panneau de navigation.
2. Dans le champ correspondant à la IISExamplecouche, choisissez Recipes.
3. Dans le coin supérieur droit, choisissez Edit (Modifier).
4. Sous Custom Chef Recipes (Recettes Chef personnalisées), dans la zone des recettes Setup (Installer), tapez **`iis-cookbook::install`**.

   > **Note**
   >
   > Utilisez *`cookbook-name`*`::`*`recipe-name`* pour identifier les recettes, où vous omettez le suffixe `.rb` du nom de la recette.

5. Choisissez + pour ajouter la recette à la couche. Un x rouge s'affiche à côté de la recette pour faciliter sa suppression ultérieurement.
6. Choisissez Save (Enregistrer) pour enregistrer la nouvelle configuration. Les recettes Setup personnalisées doivent désormais inclure `iis-cookbook::install`.

Ajouter une instance à la couche et la démarrer

Vous pouvez essayer la recette en ajoutant une instance à la couche et en démarrant l'instance. AWS OpsWorks Stacks installe automatiquement les livres de recettes et s'exécute `install.rb` pendant l'installation, dès que l'instance a fini de démarrer.

Pour ajouter une instance à une couche et la démarrer

1. Dans le volet de navigation AWS OpsWorks Stacks, sélectionnez Instances.
2. Sous IISExamplecouche, choisissez Ajouter une instance.
3. Sélectionnez la taille appropriée. t2.micro (ou la plus petite taille à votre disposition) doit suffire pour l'exemple.
4. Choisissez Add Instance (Ajouter une instance). Par défaut, AWS OpsWorks Stacks génère des noms d'instance en ajoutant un entier au nom abrégé de la couche. L'instance doit donc être nommée iisexample1.
5. Choisissez démarrer dans la colonne Actions de l'instance pour démarrer l'instance. AWS OpsWorks Stacks lancera ensuite une EC2 instance et exécutera les recettes d'installation pour la configurer. Si la couche avait des recettes de déploiement à ce stade, AWS OpsWorks Stacks les exécuterait une fois les recettes de configuration terminées.

   Le processus peut prendre quelques minutes, pendant lesquelles la colonne Status (Statut) affiche une série d'états de statuts. Lorsque vous parvenez au statut online (en ligne), le processus d'installation est terminé et l'instance est prête à être utilisée.

Confirmer qu'IIS est installé et en cours d'exécution

Vous pouvez utiliser RDP pour vous connecter à l'instance et vérifier que votre recette Setup a fonctionné correctement.

Pour vérifier qu'IIS est installé et en cours d'exécution

1. Choisissez Instances dans le volet de navigation et choisissez rdp dans la colonne Actions de l'instance iisexample1. AWS OpsWorks Stacks génère automatiquement pour vous un mot de passe RDP qui expire après une période spécifiée.
2. Définissez Session valid for (Session valide pour) sur 2 heures et choisissez Generate Password (Générer un mot de passe).
3. AWS OpsWorks Stacks affiche le mot de passe ainsi que, pour votre commodité, le nom DNS public et le nom d'utilisateur de l'instance. Copiez les trois et cliquez sur Acknowledge and close (Accepter et fermer).
4. Ouvrez votre client RDP et utilisez les données de l'étape 3 pour vous connecter à l'instance.
5. Sur l'instance, ouvrez l'Explorateur Windows et examinez le lecteur `C:`. Il doit avoir un répertoire `C:\inetpub`, qui a été créé par l'installation IIS.

6. Ouvrez l'application Outils d'administration du Panneau de configuration, puis ouvrez Services. Vous devez voir le service IIS vers le bas de la liste. Il se nomme World Wide Web Publishing Service et l'état doit être running (en cours d'exécution).
7. Retournez à la console AWS OpsWorks Stacks et choisissez l'adresse IP publique de l'instance iisexample1. Assurez-vous de le faire dans AWS OpsWorks Stacks, et non dans la EC2 console Amazon. Une requête HTTP est automatiquement envoyée à l'adresse, ce qui doit ouvrir la page d'accueil IIS par défaut.

La rubrique suivante explique comment déployer une application sur l'instance, une simple page HTML statique dans cet exemple. Toutefois, si vous souhaitez faire une pause, choisissez stop dans la colonne Actions de l'instance iisexample1 pour arrêter l'instance et éviter d'encourir des frais inutiles. Vous pouvez redémarrer l'instance lorsque vous êtes prêt à continuer.

Étape 2.5 : Déployer une application

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

L'installation IIS crée un répertoire `C:\inetpub\wwwroot` pour le code de votre application et les fichiers associés. L'étape suivante consiste à installer une application dans ce répertoire. Pour cet exemple, vous allez installer une page d'accueil HTML statique, `default.html`, dans `C:\inetpub\wwwroot`. Vous pouvez facilement étendre l'approche générale pour gérer des scénarios plus complexes, telles que les applications ASP.NET.

Vous pouvez inclure les fichiers de l'application dans votre livre de recettes et faire en sorte qu'`install.rb` les copie dans `C:\inetpub\wwwroot`. Pour obtenir un exemple de la façon de procéder, consultez Exemple 6 : Création de fichiers. Cependant, cette approche n'est ni très flexible ni très efficace, et il est généralement préférable de séparer le développement du livre de recettes du développement de l'application.

La solution préférée consiste à implémenter une recette de déploiement distincte qui récupère le code de l'application et les fichiers associés d'un référentiel (le référentiel de votre choix, pas

seulement le référentiel de livres de recettes) et l'installe sur chaque instance de serveur IIS. Cette approche sépare le développement de livres de recettes du développement d'applications et, lorsque vous avez besoin de mettre à jour votre application, elle vous permet simplement d'exécuter à nouveau la recette de déploiement sans avoir besoin de mettre à jour vos livres de recettes.

Cette rubrique montre comment implémenter une recette de déploiement simple qui déploie `default.htm` sur votre serveur IIS. Vous pouvez facilement étendre cet exemple à des applications plus complexes.

Rubriques

- Créer l'application et la stocker dans un référentiel
- Implémenter une recette pour déployer l'application
- Mettre à jour les livres de recette de l'instance
- Ajouter la recette à la couche IIS personnalisée
- Ajouter une application
- Déployer l'application et ouvrir l'application

Créer l'application et la stocker dans un référentiel

Vous pouvez utiliser n'importe quel référentiel de votre choix pour vos applications. Pour plus de simplicité, cet exemple stocke `default.htm` dans un compartiment S3 public.

Pour créer l'application

1. Créez un répertoire nommé `iis-application` sur un emplacement approprié de votre ordinateur.
2. Ajoutez un fichier `default.htm` à `iis-application` avec le contenu suivant.

```
<!DOCTYPE html>
<html>
  <head>
    <title>IIS Example</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

3. Créez un compartiment S3, chargez `default.htm` dans le compartiment et enregistrez l'URL en vue d'une utilisation ultérieure. Pour des raisons de simplicité, rendez le fichier public.

> **Note**
>
> Il s'agit d'une application très simple, mais vous pouvez étendre les principes de base pour gérer les applications de niveau production.
>
> - Pour les applications plus complexes avec plusieurs fichiers, il est généralement plus simple de créer une archive .zip d'`iis-application` et de la charger dans votre compartiment S3.
>
> Vous pouvez ensuite télécharger le fichier et extraire le contenu dans le répertoire approprié. Il n'y a pas besoin de télécharger plusieurs fichiers, de créer une structure de répertoire, etc.
> - Pour une application en production, vous devrez probablement conserver vos fichiers privés. Pour obtenir un exemple de la façon dont une recette télécharge les fichiers à partir d'un compartiment S3 privé, consultez Utilisation du SDK pour Ruby sur AWS OpsWorks une instance Stacks Windows.
> - Vous pouvez stocker votre application dans n'importe quel référentiel approprié.
>
> Généralement, vous téléchargez l'application à l'aide de l'API publique d'un référentiel. Cet exemple utilise l'API Amazon S3. Si, par exemple, vous stockez votre application sur GitHub, vous pouvez utiliser l'GitHub API.

Implémenter une recette pour déployer l'application

Ajoutez une recette nommée `deploy.rb` au répertoire `iis-cookbook` `recipes` avec le contenu suivant.

```
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
```

Cet exemple utilise le SDK pour Ruby v2 pour télécharger le fichier. Cependant, AWS OpsWorks Stacks n'installe pas ce SDK sur les instances Windows. La recette commence donc par la `chef_gem`ressource, qui gère cette tâche.

> **Note**
>
> La ressource `chef_gem` installe les GEM dans la version Ruby dédiée de Chef, qui est la version utilisée par les recettes. Si vous souhaitez installer un GEM pour une version Ruby à l'échelle du système, utilisez la ressource gem_package.

La majeure partie de la recette est une `ruby_block`ressource qui exécute un bloc de code Ruby qui utilise le SDK pour Ruby pour `default.htm` le téléchargement. Le code de `ruby_block` peut être réparti dans les sections suivantes, qui correspondent aux commentaires numérotés de l'exemple de code.

1 : Spécification d'un ensemble de certificats

Amazon S3 utilise le protocole SSL. Vous avez donc besoin d'un certificat approprié pour télécharger des objets depuis un compartiment S3. Le SDK pour Ruby v2 n'inclut pas de bundle

de certificats, vous devez donc en fournir un et configurer le SDK pour Ruby pour l'utiliser. AWS OpsWorks Stacks n'installe pas directement un bundle de certificats, mais il installe Git, qui inclut un bundle de certificats (`curl-ca-bundle.crt`). Pour plus de commodité, cet exemple configure le SDK pour Ruby afin d'utiliser le bundle de certificats Git pour SSL. Vous pouvez également installer votre propre ensemble et configurer le kit de développement logiciel SDK en conséquence.

2 : Récupération des données du référentiel

Pour télécharger un objet depuis Amazon S3, vous avez besoin de la région AWS, du nom du compartiment et du nom de la clé. Comme décrit plus tard, cet exemple fournit les informations en associant un ensemble de variables d'environnement à l'application. Lorsque vous déployez une application, AWS OpsWorks Stacks ajoute un ensemble d'attributs à l'objet nœud de l'instance. Ces attributs sont essentiellement une table de hachage qui contient la configuration de l'application, y compris les variables d'environnement. Les attributs d'application de cette application se présentent comme suit, au format JSON.

```
{
  "app_id": "8f71a9b5-de7f-451c-8505-3f35086e5bb3",
  "app_source": {
      "password": null,
      "revision": null,
      "ssh_key": null,
      "type": "other",
      "url": null,
      "user": null
  },
  "attributes": {
      "auto_bundle_on_deploy": true,
      "aws_flow_ruby_settings": {},
      "document_root": null,
      "rails_env": null
  },
  "data_sources": [{"type": "None"}],
  "domains": ["iis_example_app"],
  "enable_ssl": false,
  "environment": {
      "S3REGION": "us-west-2",
      "BUCKET": "windows-example-app",
      "FILENAME": "default.htm"
  },
  "name": "IIS-Example-App",
```

```
  "shortname": "iis_example_app",
  "ssl_configuration": {
      "certificate": null,
      "private_key": null,
      "chain": null
  },
  "type": "other",
  "deploy": true
}
```

Les variables d'environnement de l'application sont stockées dans l'attribut `[:environment]`. Pour les récupérer, utilisez une requête de recherche Chef et extrayez la table de hachage de l'application ; qui se trouve sous le nœud `aws_opsworks_app`. Comme cette application sera définie comme ayant le type `other`, la requête recherche les applications de ce type. Comme la recette tire parti du fait qu'il n'y a qu'une seule application sur cette instance, la table de hachage digne d'intérêt est juste `app[0]`. Pour plus de commodité, la recette attribue ensuite les noms de région, de compartiment et de fichier à des variables.

Pour plus d'informations sur l'utilisation de la recherche Chef, consultez Obtention des valeurs d'attribut avec la recherche de Chef.

3 : Téléchargement du fichier

La troisième partie de la recette crée un objet client S3 et utilise sa méthode `get_object` pour télécharger `default.htm` dans le répertoire `C:\inetpub\wwwroot` de l'instance.

> **Note**
>
> Comme une recette est une application Ruby, le code Ruby ne doit pas nécessairement être dans un `ruby_block`. Cependant, le code du corps de la recette s'exécute en premier, suivi, dans l'ordre, par les ressources. Dans cet exemple, si vous insérez le code de téléchargement dans le corps de la recette, cela échouera car la `chef_gem` ressource n'aurait pas encore installé le SDK pour Ruby. Le code de la `ruby_block` ressource s'exécute lorsque la ressource s'exécute, une fois que la `chef_gem` ressource a installé le SDK pour Ruby.

Mettre à jour les livres de recette de l'instance

AWS OpsWorks Stacks installe automatiquement des livres de recettes personnalisés sur les nouvelles instances. Cependant, comme vous travaillez avec une instance existante, vous devez mettre à jour votre livre de recettes manuellement.

Pour mettre à jour les livres de recette de l'instance

1. Créez une archive `.zip` d'`iis-cookbook` et chargez-la dans le compartiment S3.

   Le livre de recettes existant est remplacé, mais comme l'URL demeure la même, vous n'avez pas besoin de mettre à jour la configuration de la pile.
2. Si votre instance n'est pas en ligne, redémarrez-la.
3. Une fois que l'instance est en ligne, choisissez Stack (Pile) dans le panneau de navigation, puis Run Command (Exécuter une commande).
4. Pour Command (Commande), choisissez Update Custom Cookbooks (Mettre à jour les livres de recettes personnalisées). Cette commande installe le livre de recettes mis à jour sur l'instance.
5. Choisissez Update Custom Cookbooks (Mettre à jour les livres de recettes personnalisées). L'exécution de la commande peut prendre quelques minutes.

Ajouter la recette à la couche IIS personnalisée

Comme avec `install.rb`, le meilleur moyen de gérer le déploiement consiste à attribuer `deploy.rb` à l'événement du cycle de vie approprié. Généralement, vous assignez les recettes du déploiement à l'événement Deploy et celles-ci sont appelées collectivement recettes Deploy. L'affectation d'une recette à l'événement Deploy ne déclenche pas l'événement. Au lieu de cela :

- Pour les nouvelles instances, AWS OpsWorks Stacks exécute automatiquement les recettes de déploiement une fois les recettes de configuration terminées, de sorte que les nouvelles instances disposent automatiquement de la version actuelle de l'application.
- Dans le cas des instances en ligne, vous utilisez une commande deploy pour installer manuellement les applications nouvelles ou mises à jour.

  Cette commande déclenche un événement Deploy sur les instances de la pile, ce qui exécute les recettes Deploy.

Pour attribuer deploy.rb à l'événement Deploy de la couche

1. Choisissez Layers dans le volet de navigation, puis choisissez Recipes sous Layer IISExample.
2. Sous Custom Chef Recipes (Recettes Chef personnalisées), ajoutez **`iis-cookbook::deploy`** à la zone des recettes Deploy et choisissez + pour ajouter la recette à la couche.
3. Choisissez Save (Enregistrer) pour enregistrer la nouvelle configuration. Les recettes Deploy personnalisées doivent désormais inclure `iis-cookbook::deploy`.

Ajouter une application

La dernière tâche consiste à ajouter une application à la pile pour représenter votre application dans l'environnement AWS OpsWorks Stacks. Une application inclut les métadonnées telles que le nom complet de l'application et les données requises pour télécharger l'application à partir de son référentiel.

Pour ajouter l'application à la pile

1. Choisissez Apps (Applications) dans le panneau de navigation, puis choisissez Add an app (Ajouter une application).
2. Configurez l'application avec les paramètres suivants.
   - Nom — l **`IIS-Example-App`**
   - Type de référentiel — Autre
   - Variables d'environnement — Ajoutez les trois variables d'environnement suivantes :
     - **`S3REGION`**— La région du compartiment (dans ce cas,`us-west-1`).
     - **`BUCKET`**— Le nom du bucket, tel que`windows-example-app`.
     - **`FILENAME`**— Le nom du fichier :**`default.htm`**.
3. Acceptez les valeurs par défaut pour les paramètres restants, puis choisissez Add App (Ajouter une application) pour ajouter l'application à la pile.

> **Note**
>
> Cet exemple utilise des variables d'environnement pour fournir les données de téléchargement. Une autre approche consiste à utiliser un type de référentiel S3 Archive et à fournir l'URL du fichier. AWS OpsWorks Stacks ajoute les informations, ainsi que des données facultatives, telles que vos informations d'identification AWS, à l'`app_source`attribut

de l'application. Votre recette Deploy doit récupérer l'URL à partir des attributs de l'application et l'analyser pour extraire la région, le nom du compartiment et le nom du fichier.

Déployer l'application et ouvrir l'application

AWS OpsWorks Stacks déploie automatiquement les applications sur les nouvelles instances, mais pas sur les instances en ligne. Comme votre instance est déjà en cours d'exécution, vous devez déployer l'application manuellement.

Pour déployer l'application

1. Choisissez Apps (Applications) dans le panneau de navigation, puis deploy (déployer) dans la colonne Actions de l'application.
2. Command (Commande) doit être défini sur Deploy (Déployer). Choisissez Deploy (Déployer) dans le coin inférieur droit de la page Deploy App (Déployer l'application). L'exécution de la commande peut prendre quelques minutes.

   Une fois que le déploiement est terminé, vous revenez à la page Apps (Applications). L'indicateur Status (Statut) affiche successful (succès) en vert et le nom de l'application a une coche verte à côté pour indiquer un déploiement réussi.

> **Note**
>
> Les applications Windows étant toujours de type d'application Other, le déploiement de l'application effectue ce qui suit :
>
> - Ajoute les données de l'application aux attributs de configuration et de déploiement de pile, comme indiqué plus tôt.
> - Déclenche un événement Deploy sur les instances de la pile, ce qui exécute vos recettes Deploy personnalisées.

> **Note**
>
> Pour plus d'informations sur le dépannage des applications ou des déploiements ayant échoué, consultez Débogage des recettes.

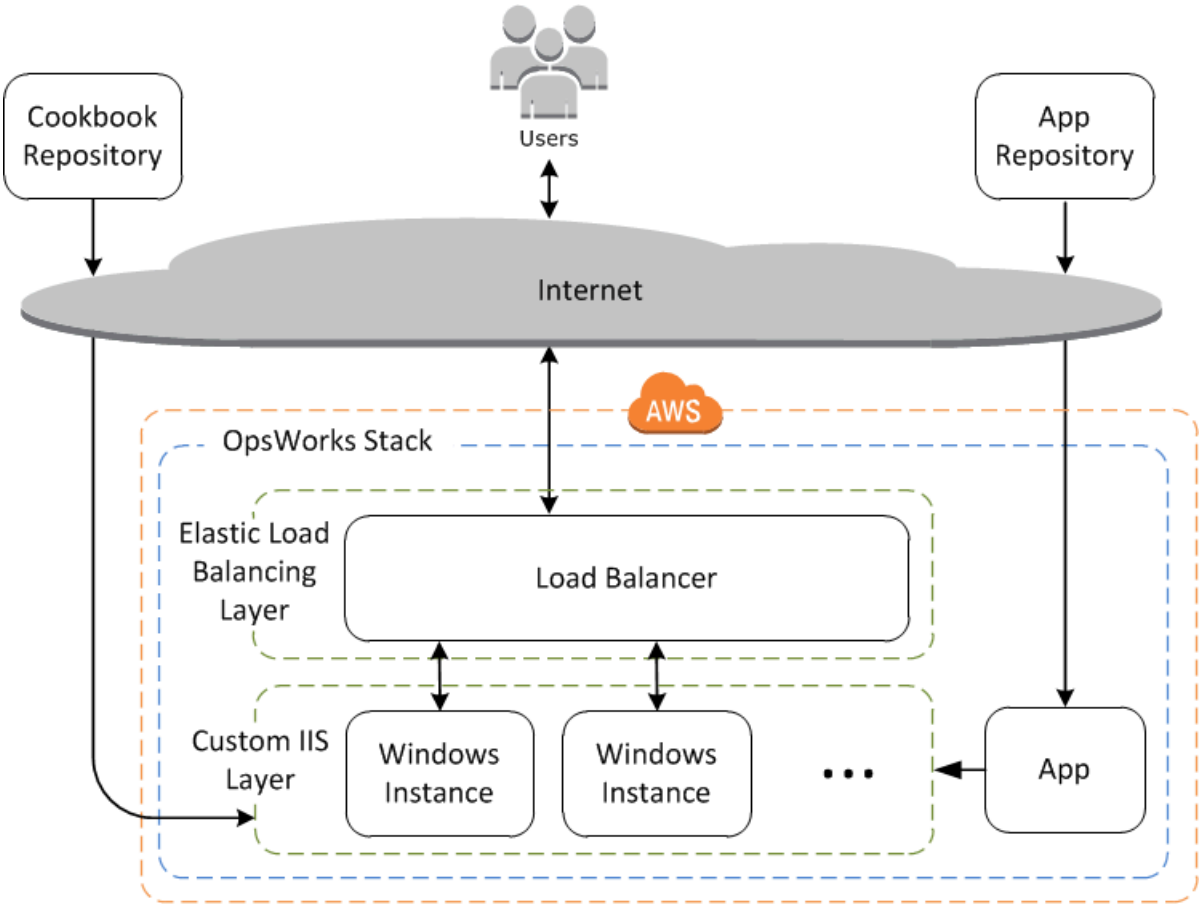
L'application est maintenant installée. Vous pouvez l'ouvrir en choisissant Instances dans le volet Navigation, puis en choisissant l'adresse IP publique de l'instance. Cela envoie une demande HTTP à l'instance et vous devriez voir quelque chose comme ce qui suit s'afficher dans votre navigateur.



Étape 3 : Extensification IISExample

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Si les demandes utilisateur entrantes approchent de la limite de ce que vous pouvez gérer avec une seule instance t2.micro, vous devrez accroître la capacité de votre serveur. Vous pouvez passer à une instance plus grande, mais cela présente des limites. Une approche plus souple consiste à ajouter des instances à votre pile et à les placer derrière un équilibreur de charge. L'architecture de base ressemble à ce qui suit.



Parmi les autres avantages, cette approche est beaucoup plus robuste qu'une seule instance volumineuse.

- Si l'une de vos instances échoue, l'équilibreur de charge répartit les demandes entrantes sur les autres instances et votre application continue à fonctionner.
- Si vous placez les instances dans différentes zones de disponibilité (pratique recommandée), votre application continue à fonctionner même si une zone de disponibilité rencontre des problèmes.

AWS OpsWorks Stacks permet de redimensionner facilement les piles. Cette section décrit les principes de base de la mise à l'échelle d'une pile en ajoutant une deuxième instance PHP App Server 24 h/24, 7 j/7 IISExample et en plaçant les deux instances derrière un équilibreur de charge Elastic Load Balancing. Vous pouvez facilement étendre la procédure pour ajouter un nombre arbitraire d'instances 24 heures sur 24, 7 jours sur 7, ou vous pouvez utiliser des instances temporelles pour que AWS OpsWorks Stacks redimensionne automatiquement votre stack. Pour de plus amples informations, veuillez consulter Gestion de la charge avec des instances basées sur le temps et sur la charge.

Ajouter un équilibreur de charge

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Elastic Load Balancing est un service AWS qui distribue automatiquement le trafic applicatif entrant sur plusieurs EC2 instances Amazon. Un équilibreur de charge peut obéir à deux objectifs. L'objectif manifeste consiste à équilibrer la charge sur vos serveurs d'applications. La plupart des sites préfèrent isoler leurs serveurs d'applications et bases de données d'un accès direct de l'utilisateur. Outre la distribution du trafic, Elastic Load Balancing effectue les tâches suivantes :

- Détecte les EC2 instances Amazon défectueuses.

  Il redirige le trafic vers les instances saines restantes en attendant que les instances défectueuses soient restaurées.
- Dimensionne automatiquement la capacité de traitement des demandes en réponse au trafic entrant.

> **Note**
>
> AWS OpsWorks Stacks ne prend pas en charge Application Load Balancer. Vous ne pouvez utiliser Classic Load Balancer qu'avec AWS OpsWorks Stacks.

Bien qu'Elastic Load Balancing soit souvent désigné sous le nom de couche, il fonctionne un peu différemment des autres couches intégrées. Au lieu de créer une couche et d'y ajouter des instances, vous créez un équilibreur de charge Elastic Load Balancing à l'aide de la EC2 console Amazon, puis vous l'attachez à l'une de vos couches existantes, généralement une couche de serveur d'applications. AWS OpsWorks Stacks enregistre ensuite les instances existantes de la couche auprès du service et ajoute automatiquement les nouvelles instances. La procédure suivante décrit comment ajouter un équilibreur de charge.

Pour attacher un équilibreur de charge à la couche IIS personnalisée

1. Utilisez la EC2 console Amazon pour créer un nouvel équilibreur de charge pour IISExample. Pour plus d'informations, consultez Mise en route avec Elastic Load Balancing. Lorsque vous exécutez l'Assistant Créer un équilibreur de charge, configurez l'équilibreur de charge comme suit :

   1 : Définir l'équilibreur de charge

   Attribuez à l'équilibreur de charge un nom facilement reconnaissable, tel que IIS-LB, afin de le localiser plus facilement dans la AWS OpsWorks console Stacks. Acceptez les valeurs par défaut pour les paramètres restants, puis choisissez Next: Assign Security Groups (Suivant : Attribuer les groupes de sécurité).

   2 : Attribuer les groupes de sécurité

   Si votre compte prend en charge le VPC par défaut, l'Assistant affiche cette page pour déterminer le groupe de sécurité de l'équilibreur de charge. Cette page n'est pas affichée pour EC2 Classic.

   Pour cette procédure pas à pas, spécifiez default VPC security group (groupe de sécurité VPC par défaut), puis choisissez Next: Configure Security Settings (Suivant : Configurer les paramètres de sécurité).

   3 : Configurer les paramètres de sécurité

   Comme cette procédure requiert que votre équilibreur de charge utilise un auditeur sécurisé (à savoir, HTTPS ou SSL sur sa connexion frontale), choisissez Next: Configure Health Check (Suivant : Configurer la vérification de l'état) pour continuer.

   4 : Configurer la vérification de l'état

   Définissez le chemin ping sur **/**. Acceptez les valeurs par défaut pour les autres paramètres, puis choisissez Next : Add EC2 Instances.

   5 : Ajouter des EC2 instances

   AWS OpsWorks Stacks se charge automatiquement de l'enregistrement des instances auprès de l'équilibreur de charge. Choisissez Next: Add Tags (Suivant : Ajouter des balises) pour continuer.

6 : Ajouter des balises

Vous n'utiliserez pas de balises pour cet exemple. Choisissez Review and Create (Vérifier et créer).

7 : Vérifier

Vérifiez vos choix et sélectionnez Create (Créer), puis Close (Fermer), qui lance l'équilibreur de charge.

2. Si votre compte prend en charge les VPC par défaut, après que vous avez démarré l'équilibreur de charge, vous devez vous assurer que son groupe de sécurité a les règles entrantes appropriées. La règle par défaut n'accepte pas de trafic entrant.

   1. Choisissez Security Groups dans le volet EC2 de navigation Amazon.
   2. Choisissez default VPC security group (groupe de sécurité VPC par défaut)
   3. Dans l'onglet Entrant, choisissez Modifier.
   4. Pour cette procédure pas à pas, définissez Source sur N'importe où, ce qui demande à l'équilibreur de charge d'accepter le trafic entrant à partir de n'importe quelle adresse IP.
   5. Cliquez sur Sauvegarder

3. Retournez à la console AWS OpsWorks Stacks. Sur la page Layers (Couches), choisissez Network (Réseau).
4. Sous Elastic Load Balancing, sélectionnez l'équilibreur de charge IIS-LB que vous avez créé à l'étape 1, puis cliquez sur Save (Enregistrer).

   Une fois que vous avez attaché l'équilibreur de charge à la couche, AWS OpsWorks Stacks enregistre automatiquement les instances actuelles de la couche et en ajoute de nouvelles au fur et à mesure de leur mise en ligne.
5. Sur la page Couches, cliquez sur le nom de l'équilibreur de charge pour ouvrir la page des détails. Une coche verte à côté de l'instance sur la page de l'équilibreur de charge indique que l'instance a passé avec succès une vérification du statut.

Vous pouvez désormais exécuter IIS-Example-App en envoyant une demande à l'équilibreur de charge.

Pour exécuter l' IIS-Example-Appéquilibreur de charge

1. Choisissez Layers (Couches). L'équilibreur de charge IIS-ELB doit être répertorié comme couche et la colonne Health doit avoir une seule instance en vert, ce qui indique une instance saine.
2. Choisissez le nom DNS de l'équilibreur de charge à exécuter IIS-Example-App. Il doit figurer sous le nom de l'équilibreur de charge et se présenter sous la forme `IIS-LB-1802910859.us-west-2.elb.amazonaws.com`. L'équilibreur de charge achemine la demande vers l'instance et retourne la réponse, qui doit être exactement identique à celle que vous obtenez lorsque vous cliquez sur l'adresse IP publique de l'instance.

Comme vous avez une seule instance à ce stade, l'équilibreur de charge n'ajoute pas vraiment grand chose. Cependant, vous pouvez maintenant ajouter des instances supplémentaires à la couche.

Pour ajouter une instance à la couche

1. Choisissez Instances, puis + instance pour ajouter une autre instance à la couche.
2. Démarrez l'instance.

Comme il s'agit de nouvelles instances, AWS OpsWorks Stacks installe automatiquement les livres de recettes personnalisés actuels et déploie la version actuelle de l'application lors de la configuration. Lorsque l'instance est en ligne, AWS OpsWorks Stacks l'ajoute automatiquement à l'équilibreur de charge, de sorte que votre instance commence immédiatement à traiter les demandes. Pour vérifier que l'application fonctionne encore, vous pouvez choisir à nouveau le nom DNS de l'équilibreur de charge.

Étapes suivantes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette procédure pas à pas vous a guidé à travers les bases de l'installation d'une simple pile serveur d'applications Windows. Voici quelques suggestions sur ce qu'il convient de faire ensuite.

- Si vous souhaitez en savoir plus, Mise en route : livres de recettes propose un didacticiel d'introduction à la mise en œuvre de livres de recettes et inclut un certain nombre d'exemples spécifiques à AWS OpsWorks Stacks.
- Vous pouvez ajouter une couche Amazon Relational Database Service (Amazon RDS) à la pile pour l'utiliser comme serveur de base de données principal. Pour plus d'informations sur la connexion de votre application à la base de données, consultez Utilisation d'une recette personnalisée.

Commencer à utiliser Cookbooks in Stacks AWS OpsWorks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une pile AWS OpsWorks Stacks au niveau de la production nécessite généralement une certaine personnalisation, ce qui implique souvent la mise en œuvre d'un livre de recettes Chef personnalisé. Un livre de recettes est un fichier de package qui contient les informations de configuration, y compris les instructions appelées recettes. Une recette est un ensemble d'une ou plusieurs instructions, écrites dans la syntaxe du langage Ruby, qui spécifie les ressources à utiliser et l'ordre dans lequel ces ressources sont appliquées. Une ressource, telle qu'utilisée dans Chef, est une déclaration de stratégie de configuration. Cette procédure pas à pas fournit une introduction de base à la mise en œuvre des livres de recettes Chef pour AWS OpsWorks Stacks. Pour en savoir plus sur les ressources, livres de cuisine et recettes Chef, consultez les liens dans Étapes suivantes.

Cette procédure pas à pas décrit principalement comment créer vos propres livres de recettes. Vous pouvez également utiliser les livres de recettes mis à disposition par la communauté sur des sites Web tels que Chef Supermarket. Pour vous aider à vous familiariser avec les livres de recettes de la communauté, nous incluons les instructions d'utilisation d'un livre de recettes de la communauté du Chef Supermarket plus loin dans la procédure pas à pas.

Avant de commencer cette procédure, procédez à quelques étapes de configuration. Si vous avez déjà terminé l'une des autres procédures pas à pas de ce chapitre, telle que Mise en route : exemple, vous remplissez les conditions requises pour cette procédure pas à pas et pouvez directement démarrer la procédure pas à pas. Sinon, veillez à prendre en compte les prérequis, puis revenez à cette procédure pas à pas.

Rubriques

- Étape 1 : Créer le livre de recettes
- Étape 2 : Créer la pile et ses composants
- Étape 3 : Exécuter et tester la recette
- Étape 4 : Mettre à jour le livre de recettes pour installer un package
- Étape 5 : Mettre à jour le livre de recettes sur l'instance et exécuter la recette
- Étape 6 : Mettre à jour le livre de recettes pour ajouter un utilisateur
- Étape 7 : Mettre à jour le livre de recettes pour créer un répertoire
- Étape 8 : Mettre à jour le livre de recettes pour créer et copier des fichiers
- Étape 9 : Mettre à jour le livre de recettes pour exécuter une commande
- Étape 10 : Mettre à jour le livre de recettes pour exécuter un script
- Étape 11 : Mettre à jour le livre de recettes pour gérer un service
- Étape 12 : Mettre à jour le livre de recettes pour utiliser JSON personnalisé
- Étape 13 : Mettre à jour le livre de recettes pour utiliser les conteneurs de données
- Étape 14 : Mettre à jour le livre de recettes pour utiliser l'itération
- Étape 15 : Mettre à jour le livre de recettes pour utiliser la logique conditionnelle
- Étape 16 : Mettre à jour le livre de recettes pour utiliser les livres de recettes de la communauté
- Étape 17 : (Facultatif) Nettoyer
- Étapes suivantes

Étape 1 : Créer le livre de recettes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement

aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Commencez par créer un livre de recettes. Ce livre de recettes ne propose pas grand chose au départ, mais il sert de base pour le reste de cette procédure pas à pas.

> **Note**
>
> Cette étape explique comment créer un livre de recettes manuellement. Vous pouvez créer un livre de recettes plus rapidement avec le kit de développement Chef (Chef DK) en exécutant la commande chef generate cookbook sur votre poste de travail local. Cependant, cette commande crée plusieurs dossiers et fichiers dont vous n'aurez pas besoin pour cette procédure.

Pour créer le livre de recettes

1. Sur votre station de travail locale, créez un répertoire nommé `opsworks_cookbook_demo`. Vous pouvez utiliser un autre nom, mais n'oubliez pas de remplacer alors `opsworks_cookbook_demo` tout au long de la procédure pas à pas.
2. Dans le répertoire `opsworks_cookbook_demo`, créez un fichier nommé `metadata.rb` à l'aide d'un éditeur de texte. Ajoutez le code suivant pour indiquer le nom du livre de recettes. Pour plus d'informations sur `metadata.rb`, consultez metadata.rb sur le site web de Chef.

   ```
   name "opsworks_cookbook_demo"
   ```
3. Dans le répertoire `opsworks_cookbook_demo`, créez un sous-répertoire appelé `recipes`. Ce sous-répertoire contient toutes les recettes que vous créez pour le livre de recettes de cette procédure pas à pas.
4. Dans le répertoire `recipes`, créez un fichier nommé `default.rb`. Ce fichier contient une recette avec le même nom que le fichier, mais sans l'extension de fichier : `default`. Ajoutez la ligne de code suivante au fichier `default.rb`. Ce code est une recette d'une seule ligne qui affiche un simple message dans le journal lors de l'exécution de la recette :

   ```
   Chef::Log.info("********** Hello, World! **********")
   ```

5. Depuis le terminal ou l'invite de commande, utilisez la commande tar pour créer un fichier nommé `opsworks_cookbook_demo.tar.gz`, qui contient le répertoire `opsworks_cookbook_demo` et son contenu. Par exemple :

   ```
   tar -czvf opsworks_cookbook_demo.tar.gz opsworks_cookbook_demo/
   ```

   Vous pouvez utiliser un autre nom de fichier, mais n'oubliez pas de remplacer alors `opsworks_cookbook_demo.tar.gz` tout au long de la procédure pas à pas.

   > **Note**
   >
   > Lorsque vous créez le fichier `tar` sous Windows, le répertoire supérieur doit être le répertoire parent du livre de recettes. Cette procédure pas à pas a été testée sur Linux avec la commande tar fournie par le package `tar` et sur Windows avec la commande tar fournie par Git Bash. L'utilisation d'autres commandes ou programmes pour créer un fichier TAR compressé (.tar.gz) peut ne pas fonctionner comme prévu.

6. Créez un compartiment S3 ou utilisez un compartiment existant. Pour plus d'informations, consultez Créer un compartiment.
7. Chargez le fichier `opsworks_cookbook_demo.tar.gz` dans le compartiment S3. Pour plus d'informations, consultez Ajouter un objet à un compartiment.

Vous avez maintenant un livre de recettes que vous utiliserez tout au long de cette procédure pas à pas.

À l'étape suivante, vous créez une pile AWS OpsWorks Stacks que vous utiliserez ultérieurement pour télécharger votre livre de recettes et pour exécuter les recettes du livre de recettes.

Étape 2 : Créer la pile et ses composants

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Créez une pile AWS OpsWorks Stacks et ses composants, notamment une couche et une instance. Dans les étapes ultérieures, vous chargez votre livre de recettes sur l'instance, puis exécutez les recettes du livre de recettes sur cette instance.

Pour créer la pile

1. Connectez-vous à la console AWS OpsWorks Stacks sur https://console.aws.amazon.com/opsworks.
2. Exécutez l'une des actions suivantes, si elles s'appliquent :
   - Si la page Bienvenue dans AWS OpsWorks Stacks s'affiche, choisissez Ajouter votre première pile ou Ajouter votre première pile AWS OpsWorks Stacks (les deux choix ont le même effet). La page Add stack (Ajouter une pile) s'affiche.
   - Si la page OpsWorks Tableau de bord s'affiche, choisissez Ajouter une pile. La page Add stack (Ajouter une pile) s'affiche.
3. Choisissez Chef 12 stack (Pile Chef 12).
4. Dans la zone Stack name (Nom de pile), tapez le nom de la pile (par exemple **`MyCookbooksDemoStack`**). Vous pouvez saisir un autre nom, mais n'oubliez pas de remplacer `MyCookbooksDemoStack` tout au long de la procédure pas à pas.
5. Pour Région, choisissez USA West (Oregon).
6. Pour VPC, effectuez l'une des actions suivantes :
   - Si un VPC est disponible, choisissez-le. Pour de plus amples informations, veuillez consulter Running a Stack in a VPC.
   - Sinon, choisissez No VPC.
7. Pour Use custom Chef cookbooks (Utiliser les livres de recettes Chef personnalisés), choisissez Yes (Oui).
8. Pour Repository type (Type de référentiel), choisissez S3 Archive (Archive S3).

   > **Note**
   >
   > Dans la procédure pas à pas Mise en route : Linux, vous avez choisi Http Archive (Archive HTTP). Veillez à la place à choisir ici S3 Archive (Archive S3).

9. Pour Repository URL (URL du référentiel), tapez le chemin d'accès à votre fichier `opsworks_cookbook_demo.tar.gz` dans S3. Pour obtenir le chemin, dans la console

S3, sélectionnez le fichier `opsworks_cookbook_demo.tar.gz`. Dans le volet Properties (Propriétés), copiez la valeur du champ Link (Lien). (Il doit être similaire à ceci : `https://s3.amazonaws.com/amzn-s3-demo-bucket/opsworks_cookbook_demo.tar.gz`.)

10. Si votre compartiment S3 est privé, ce qui est le cas par défaut, alors pour ID de clé d'accès et clé d'accès secrète, tapez l'ID de clé d'accès et la clé d'accès secrète de l'utilisateur IAM que vous utilisez pour cette procédure pas à pas. Pour plus d'informations, consultez Modification des autorisations d'un objet et Partager un objet avec d'autres.
11. Conservez les valeurs par défaut pour les éléments suivants :
    - Default Availability Zone (Zone de disponibilité par défaut) (us-west-2a)
    - Système d'exploitation par défaut (Linux et Amazon Linux 2016.09)
    - Default SSH key (Clé SSH par défaut) (Do not use a default SSH key [Ne pas utiliser de clé SSH par défaut])
    - Stack color (Couleur de la pile) (bleu foncé)
12. Choisir Advanced (Avancé).
13. Pour IAM role (Rôle IAM), effectuez l'une des opérations suivantes :
    - S'il aws-opsworks-service-roleest disponible, choisissez-le.
    - Si aws-opsworks-service-rolece n'est pas le cas, choisissez Nouveau rôle IAM.
14. Dans le profil d'instance IAM par défaut, effectuez l'une des opérations suivantes :
    - Si aws-opsworks-ecdeux rôles sont disponibles, choisissez-le.
    - Si aws-opsworks-ec2 rôles n'est pas disponible, choisissez Nouveau profil d'instance IAM.
15. Conservez les valeurs par défaut pour les éléments suivants :
    - Default root device type (Type du périphérique racine par défaut) (EBS backed [Basé sur EBS])
    - Hostname theme (Thème du nom d'hôte) (Layer Dependent [Dépendant de la couche])
    - OpsWorks Version de l'agent (version la plus récente)
    - Custom Chef JSON (JSON Chef personnalisé) (vide)
    - Sécurité, Utiliser des groupes OpsWorks de sécurité (Oui)
16. Choisissez Ajouter une pile. AWS OpsWorks Stacks crée la pile et affiche la MyCookbooksDemoStackpage.

Pour créer la couche

1. Dans le panneau de navigation du service, choisissez Layers (Couches). La page Layers (Couches) s'affiche.
2. Choisissez Add a layer (Ajouter une couche).
3. OpsWorksDans l'onglet, pour Nom, tapez**`MyCookbooksDemoLayer`**. Vous pouvez saisir un autre nom, mais n'oubliez pas de remplacer `MyCookbooksDemoLayer` tout au long de la procédure pas à pas.
4. Pour Short name (Nom court), tapez **`cookbooks-demo`**. Vous pouvez saisir un autre nom, mais n'oubliez pas de remplacer `cookbooks-demo` tout au long de la procédure pas à pas.
5. Choisissez Ajouter une couche. AWS OpsWorks Stacks ajoute la couche et affiche la page Couches.

Pour créer et démarrer l'instance

1. Dans le panneau de navigation du service, choisissez Instances. La page Instances s'affiche.
2. Choisissez Add an instance (Ajouter une instance).
3. Sous l'onglet New (Nouveau), choisissez Advanced (Avancé).
4. Conservez les valeurs par défaut pour les éléments suivants :
   - Hostname (Nom d'hôte) (cookbooks-demo1)
   - Size (Taille) (c3.large)
   - Sous-réseau (*`IP address`*us-west-2a)
   - Scaling type (Type de dimensionnement) (24/7)
   - SSH key (Clé SSH) (Do not use a default SSH key [Ne pas utiliser de clé SSH par défaut])
   - Système d'exploitation (Amazon Linux 2016.09)
   - OpsWorks Version de l'agent (hériter de la pile)
   - Tenancy (Location) (Default - Rely on VPC settings [Par défaut - S'appuyer sur les paramètres VPC])
   - Root device type (Type de périphérique racine) (EBS backed [Basé sur EBS])
   - Volume type (Type de volume) (General Purpose (SSD) [Usage général (SSD)])
   - Volume size (Taille du volume) (8)
5. Choisissez Add instance (Ajouter une instance).

6. Pour MyCookbooksDemoLayer, pour cookbooks-demo1, pour Actions, choisissez Démarrer. Ne poursuivez pas tant que Status (Statut) ne prend pas la valeur online (en ligne). Comme ce processus peut prendre plusieurs minutes, soyez patient.

Vous avez maintenant une pile, une couche et une instance sur lesquelles le livre de recettes a été automatiquement copié à partir de votre compartiment S3. Dans l'étape suivante, vous allez exécuter et tester la recette par défaut depuis le livre de recettes sur l'instance.

Étape 3 : Exécuter et tester la recette

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Exécutez et testez la `default` recette depuis le livre de recettes que AWS OpsWorks Stacks a copié sur l'instance. Comme vous vous en souvenez, il s'agit de la recette en une ligne qui affiche un message simple dans le journal lors de l'exécution de la recette.

Pour exécuter la recette

1. Dans le panneau de navigation du service, choisissez Stack (Pile). La page MyCookbooksDemoStack s'affiche.
2. Choisissez Run Command (Exécuter une commande). La page Run Command (Exécuter une commande) s'affiche.
3. Pour Command (Commande), choisissez Execute Recipes (Exécuter les recettes).
4. Pour Recipes to execute (Recettes à exécuter), tapez **`opsworks_cookbook_demo::default`**.

   `opsworks_cookbook_demo` est le nom du livre de recettes, tel que défini dans le fichier `metadata.rb`. **`default`** est le nom de la recette à exécuter, à savoir, le nom du fichier `default.rb` du sous-répertoire `recipes` du livre de recettes, sans l'extension de fichier.
5. Conservez les paramètres par défaut suivants :

   - Comment (Commentaire) (vide)

- Advanced (Avancé), Custom Chef JSON (JSON Chef personnalisé) (vide)
- Instances (Sélectionnez tout coché, MyCookbooksDemoLayercoché, cookbooks-demo1 coché)

6. Choisissez Execute Recipes (Exécuter les recettes). La page Running command execute_recipes (Exécution de la commande execute_recipes) s'affiche. Ne poursuivez pas tant que Status (Statut) ne prend pas la valeur successful (succès). Comme ce processus peut prendre quelques minutes, soyez patient.

Pour vérifier les résultats de la recette

1. Avec la page Running command execute_recipes (Exécution de la commande execute_recipes) affichée, pour cookbooks-demo1, pour Log (Journal), choisissez show (afficher). La page de journal execute_recipes s'affiche.
2. Faites défiler le journal et recherchez une entrée qui ressemble à ce qui suit :

```
[2015-11-13T19:14:39+00:00] INFO: ********** Hello, World! **********
```

Vous avez exécuté avec succès votre première recette ! Dans l'étape suivante, vous allez mettre à jour votre livre de recettes en ajoutant une recette qui installe un package sur l'instance.

Étape 4 : Mettre à jour le livre de recettes pour installer un package

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Mettez à jour votre livre de recettes en ajoutant une recette qui installe sur l'instance d'un package contenant le célèbre éditeur de texte GNU Emacs.

Bien que vous puissiez tout aussi bien vous connecter à l'instance et installer le package une seule fois, l'écriture d'une recette vous permet de l'exécuter une fois depuis AWS OpsWorks Stacks pour installer simultanément plusieurs packages sur plusieurs instances d'une pile.

Pour mettre à jour le livre de recettes pour installer un package

1. De retour sur votre ordinateur local, dans le sous-répertoire `recipes` du répertoire `opsworks_cookbook_demo`, créez un fichier nommé `install_package.rb` avec le code suivant :

   ```
   package "Install Emacs" do
     package_name "emacs"
   end
   ```

   Cette recette installe le package `emacs` sur l'instance. (Pour plus d'informations, consultez package.)

   > **Note**
   >
   > Vous pouvez donner à une recette le nom de votre choix. Assurez-vous simplement de spécifier le nom correct de la recette chaque fois que vous souhaitez que AWS OpsWorks Stacks exécute la recette.

2. Depuis le terminal ou l'invite de commande, utilisez la commande tar pour créer une nouvelle version du fichier `opsworks_cookbook_demo.tar.gz`, qui contient le répertoire `opsworks_cookbook_demo` et son contenu chargé.
3. Chargez le fichier `opsworks_cookbook_demo.tar.gz` mis à jour dans votre compartiment S3.

Cette nouvelle recette s'exécute lorsque vous mettez à jour le livre de recettes sur l'instance et exécutez ensuite la nouvelle recette depuis le livre de recettes mis à jour. L'étape suivante décrit comment procéder.

Une fois que vous avez terminé l'étape suivante, connectez-vous à l'instance, puis tapez emacs à partir de l'invite de commande pour lancer GNU Emacs. (Pour plus d'informations, consultez Connexion à l'instance Linux.) Pour quitter GNU Emacs, appuyez sur Ctrl+X, puis sur Ctrl+C.

> **Important**
>
> Pour vous connecter à l'instance, vous devez d'abord fournir à AWS OpsWorks Stacks des informations sur votre clé SSH publique (que vous pouvez créer à l'aide d'outils tels que ssh-keygen ou PuTTYgen), puis vous devez définir des autorisations sur la `MyCookbooksDemoStack` pile pour permettre à votre utilisateur de se connecter à l'instance. Pour obtenir les instructions, consultez Enregistrement de la clé SSH publique d'un utilisateur et Connexion avec SSH.

Étape 5 : Mettre à jour le livre de recettes sur l'instance et exécuter la recette

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Mettez à jour le livre de recettes sur l'instance, puis exécutez la recette depuis le livre de recettes mis à jour de l'instance. Dans le reste de cette procédure, vous répétez cette étape chaque fois que vous mettez à jour le livre de recettes en ajoutant une nouvelle recette.

Pour mettre à jour le livre de recettes sur l'instance

1. Dans le panneau de navigation du service, choisissez Stack (Pile). La page MyCookbooksDemoStack s'affiche.
2. Choisissez Run Command (Exécuter une commande). La page Run Command (Exécuter une commande) s'affiche.
3. Pour Command (Commande), choisissez Update Custom Cookbooks (Mettre à jour les livres de recettes personnalisées).
4. Conservez les paramètres par défaut suivants :
   - Comment (Commentaire) (vide)
   - Advanced (Avancé), Custom Chef JSON (JSON Chef personnalisé) (vide)

- Avancé, Instances (Sélectionnez tout coché, MyCookbooksDemoLayercoché, cookbooks-demo1 coché)

5. Choisissez Update Custom Cookbooks (Mettre à jour les livres de recettes personnalisées). La page Running command update_custom_cookbooks (Exécution de la commande update_custom_cookbooks) s'affiche. Ne poursuivez pas tant que Status (Statut) ne prend pas la valeur successful (succès). Comme ce processus peut prendre plusieurs minutes, soyez patient.

Pour exécuter la recette

1. Dans le panneau de navigation du service, choisissez Stack (Pile). La page MyCookbooksDemoStack s'affiche.
2. Choisissez Run Command (Exécuter une commande). La page Run Command (Exécuter une commande) s'affiche.
3. Pour Command (Commande), choisissez Execute Recipes (Exécuter les recettes).
4. Pour Recipes to execute (Recettes à exécuter), tapez le nom de la recette à exécuter. La première fois où vous procédez ainsi, la recette se nomme **`opsworks_cookbook_demo::install_package`**.

   > **Note**
   >
   > Lorsque vous répéterez cette procédure plus tard, tapez le nom du livre de recettes (**`opsworks_cookbook_demo`**), suivi de deux signes deux-points (`::`), suivi du nom de la recette (nom de fichier de la recette, sans l'extension de fichier `.rb`).

5. Conservez les paramètres par défaut suivants :

   - Comment (Commentaire) (vide)
   - Advanced (Avancé), Custom Chef JSON (JSON Chef personnalisé) (vide)
   - Instances : Sélectionnez tout (coché, MyCookbooksDemoLayercoché, cookbooks-demo1 coché)

6. Choisissez Execute Recipes (Exécuter les recettes). La page Running command execute_recipes (Exécution de la commande execute_recipes) s'affiche. Ne poursuivez pas tant que Status (Statut) ne prend pas la valeur successful (succès). Comme ce processus peut prendre quelques minutes, soyez patient.

> **Note**
>
> Vous n'avez pas à exécuter manuellement les recettes. Vous pouvez attribuer des recettes aux événements du cycle de vie d'une couche, tels que les événements de configuration et de configuration, et AWS OpsWorks Stacks exécutera ces recettes automatiquement lorsque l'événement se produira. Pour de plus amples informations, veuillez consulter AWS OpsWorks Événements de cycle de vie.

Dans l'étape suivante, vous allez mettre à jour le livre de recettes pour ajouter un utilisateur à l'instance.

Étape 6 : Mettre à jour le livre de recettes pour ajouter un utilisateur

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Mettez à jour votre livre de recettes en ajoutant une recette qui ajoute un utilisateur local à l'instance et définit le répertoire de base et le shell de l'utilisateur. Cette action est similaire à l'exécution des commandes Linux adduser ou useradd, ou de la commande Windows net user. Vous ajoutez un utilisateur local à une instance, par exemple, lorsque vous souhaitez contrôler l'accès aux fichiers et aux répertoires de l'instance.

Vous pouvez également gérer les utilisateurs sans utiliser de livres de recettes. Pour de plus amples informations, veuillez consulter Gestion des utilisateurs.

Pour mettre à jour le livre de recettes sur l'instance et exécuter la nouvelle recette

1. De retour sur votre ordinateur local, dans le sous-répertoire `recipes` du répertoire `opsworks_cookbook_demo`, créez un fichier nommé `add_user.rb` avec le code suivant (pour plus d'informations, accédez à user) :

   ```
   user "Add a user" do
   ```

```
  home "/home/jdoe"
  shell "/bin/bash"
  username "jdoe"
end
```

2. Depuis le terminal ou l'invite de commande, utilisez la commande tar pour créer une nouvelle version du fichier `opsworks_cookbook_demo.tar.gz`, qui contient le répertoire `opsworks_cookbook_demo` et son contenu chargé.
3. Chargez le fichier `opsworks_cookbook_demo.tar.gz` mis à jour dans votre compartiment S3.
4. Suivez les procédures décrites dans Étape 5 : Mettre à jour le livre de recettes sur l'instance et exécuter la recette pour mettre à jour le livre de recettes sur l'instance et exécuter la recette. Dans la procédure « Pour exécuter la recette », pour Recipes to execute (Recettes à exécuter), tapez **`opsworks_cookbook_demo::add_user`**.

Pour tester la recette

1. Connectez-vous à l'instance, si vous le n'avez pas déjà fait.
2. Depuis l'invite de commande, exécutez la commande suivante afin de confirmer que le nouvel utilisateur a été ajouté :

   ```
   grep jdoe /etc/passwd
   ```

   Des informations relatives à l'utilisateur et similaires à ce qui suit s'affichent, y compris les détails tels que le nom de l'utilisateur, le numéro d'ID, le numéro d'ID du groupe, le répertoire de base et le shell :

   ```
   jdoe:x:501:502::/home/jdoe:/bin/bash
   ```

Dans l'étape suivante, vous allez mettre à jour le livre de recettes pour créer un répertoire sur l'instance.

Étape 7 : Mettre à jour le livre de recettes pour créer un répertoire

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Mettez à jour votre livre de recettes en ajoutant une recette qui ajoute un répertoire à l'instance. Cette action est similaire à l'exécution de la commande Linux mkdir ou des commandes Windows md ou mkdir.

Pour mettre à jour le livre de recettes sur l'instance et exécuter la nouvelle recette

1. Sur votre ordinateur local, dans le sous-répertoire `recipes` du répertoire `opsworks_cookbook_demo`, créez un fichier nommé `create_directory.rb` avec le code suivant. Pour plus d'informations, consultez directory :

   ```
   directory "Create a directory" do
     group "root"
     mode "0755"
     owner "ec2-user"
     path "/tmp/create-directory-demo"
   end
   ```

2. Depuis le terminal ou l'invite de commande, utilisez la commande tar pour créer une nouvelle version du fichier `opsworks_cookbook_demo.tar.gz`, qui contient le répertoire `opsworks_cookbook_demo` et son contenu chargé.
3. Chargez le fichier `opsworks_cookbook_demo.tar.gz` mis à jour dans votre compartiment S3.
4. Suivez les procédures décrites dans Étape 5 : Mettre à jour le livre de recettes sur l'instance et exécuter la recette pour mettre à jour le livre de recettes sur l'instance et exécuter la recette. Dans la procédure « Pour exécuter la recette », pour Recipes to execute (Recettes à exécuter), tapez **`opsworks_cookbook_demo::create_directory`**.

Pour tester la recette

1. Connectez-vous à l'instance, si vous le n'avez pas déjà fait.
2. Depuis l'invite de commande, exécutez la commande suivante afin de confirmer que le nouveau répertoire a été ajouté :

   ```
   ls -la /tmp/create-directory-demo
   ```

   Des informations sur le répertoire nouvellement ajouté s'affichent, y compris les informations telles que les autorisations, le nom du propriétaire et le nom du groupe :

   ```
   drwxr-xr-x 2 ec2-user root 4096 Nov 18 00:35 .
   drwxrwxrwt 6 root     root 4096 Nov 24 18:17 ..
   ```

Dans l'étape suivante, vous allez mettre à jour le livre de recettes pour créer un fichier sur l'instance.

Étape 8 : Mettre à jour le livre de recettes pour créer et copier des fichiers

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Mettez à jour votre livre de recettes en ajoutant une recette qui ajoute deux fichiers à l'instance. La première ressource de la recette crée intégralement un fichier avec le code de la recette. Cette action est similaire à l'exécution des commandes Linux cat, echo ou touch, ou des commandes Windows echo ou fsutil. Cette technique est utile pour quelques fichiers simples ou petits. La deuxième ressource de la recette copie un fichier du livre de recettes sur un autre répertoire de l'instance. Cette action est similaire à l'exécution de la commande Linux cp ou de la commande Windows copy. Cette technique est utile dans le cas de fichiers nombreux, volumineux ou complexes.

Avant de démarrer cette étape, complétez Étape 7 : Mettre à jour le livre de recettes pour créer un répertoire pour vous assurer que répertoire parent des fichiers existe déjà.

Pour mettre à jour le livre de recettes sur l'instance et exécuter la nouvelle recette

1. Sur votre ordinateur local, dans le répertoire `opsworks_cookbook_demo`, créez un sous-répertoire nommé `files`.
2. Dans le sous-répertoire `files`, créez un fichier nommé `hello.txt` avec le texte suivant : **`Hello, World!`**
3. Dans le sous-répertoire `recipes` du répertoire `opsworks_cookbook_demo`, créez un fichier nommé `create_files.rb` avec le code suivant. Pour plus d'informations, consultez file et cookbook_file.

   ```
   file "Create a file" do
     content "<html>This is a placeholder for the home page.</html>"
     group "root"
     mode "0755"
     owner "ec2-user"
     path "/tmp/create-directory-demo/index.html"
   end

   cookbook_file "Copy a file" do
     group "root"
     mode "0755"
     owner "ec2-user"
     path "/tmp/create-directory-demo/hello.txt"
     source "hello.txt"
   end
   ```

   La ressource `file` crée un fichier dans le chemin d'accès spécifié. La ressource `cookbook_file` copie le fichier depuis le répertoire `files` que vous venez de créer dans le livre de recettes (Chef s'attend à trouver un sous-répertoire standard nommé `files` depuis lequel il peut copier les fichiers) vers un autre répertoire sur l'instance.
4. Depuis le terminal ou l'invite de commande, utilisez la commande tar pour créer une nouvelle version du fichier `opsworks_cookbook_demo.tar.gz`, qui contient le répertoire `opsworks_cookbook_demo` et son contenu chargé.
5. Chargez le fichier `opsworks_cookbook_demo.tar.gz` mis à jour dans votre compartiment S3.
6. Suivez les procédures décrites dans Étape 5 : Mettre à jour le livre de recettes sur l'instance et exécuter la recette pour mettre à jour le livre de recettes sur l'instance et exécuter la recette.

Dans la procédure « Pour exécuter la recette », pour Recipes to execute (Recettes à exécuter), tapez **`opsworks_cookbook_demo::create_files`**.

Pour tester la recette

1. Connectez-vous à l'instance, si vous le n'avez pas déjà fait.
2. A l'invite de commande, exécutez les commandes suivantes, une à la fois, afin de confirmer que les nouveaux fichiers ont été ajoutés :

```
sudo cat /tmp/create-directory-demo/index.html

sudo cat /tmp/create-directory-demo/hello.txt
```

Le contenu du fichier s'affiche :

```
<html>This is a placeholder for the home page.</html>

Hello, World!
```

Dans l'étape suivante, vous allez mettre à jour le livre de recettes pour exécuter une commande sur l'instance.

Étape 9 : Mettre à jour le livre de recettes pour exécuter une commande

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Mettez à jour votre livre de recettes en ajoutant une recette qui exécute une commande qui crée une clé SSH sur l'instance.

Pour mettre à jour le livre de recettes sur l'instance et exécuter la nouvelle recette

1. Sur votre ordinateur local, dans le sous-répertoire `recipes` du répertoire `opsworks_cookbook_demo`, créez un fichier nommé `run_command.rb` avec le code suivant. Pour plus d'informations, consultez execute.

   ```
   execute "Create an SSH key" do
     command "ssh-keygen -f /tmp/my-key -N fLyC3jbY"
   end
   ```

2. Depuis le terminal ou l'invite de commande, utilisez la commande tar pour créer une nouvelle version du fichier `opsworks_cookbook_demo.tar.gz`, qui contient le répertoire `opsworks_cookbook_demo` et son contenu chargé.
3. Chargez le fichier `opsworks_cookbook_demo.tar.gz` mis à jour dans votre compartiment S3.
4. Suivez les procédures décrites dans Étape 5 : Mettre à jour le livre de recettes sur l'instance et exécuter la recette pour mettre à jour le livre de recettes sur l'instance et exécuter la recette. Dans la procédure « Pour exécuter la recette », pour Recipes to execute (Recettes à exécuter), tapez **`opsworks_cookbook_demo::run_command`**.

Pour tester la recette

1. Connectez-vous à l'instance, si vous le n'avez pas déjà fait.
2. A l'invite de commande, exécutez les commandes suivantes, une à la fois, afin de confirmer que la clé SSH a été créée :

   ```
   sudo cat /tmp/my-key

   sudo cat /tmp/my-key.pub
   ```

   Le contenu de la clé privée SSH et de la clé publique s'affiche :

   ```
   -----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
```

Dans l'étape suivante, vous allez mettre à jour le livre de recettes pour exécuter un script sur l'instance.

Étape 10 : Mettre à jour le livre de recettes pour exécuter un script

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Mettez à jour votre livre de recettes en ajoutant une recette qui exécute un script sur l'instance. Cette recette crée un répertoire, puis un fichier dans ce répertoire. Il est plus facile d'écrire une recette pour exécuter un script contenant plusieurs commandes que d'exécuter ces commandes l'une après l'autre.

Pour mettre à jour le livre de recettes sur l'instance et exécuter la nouvelle recette

1. Sur votre ordinateur local, dans le sous-répertoire `recipes` du répertoire `opsworks_cookbook_demo`, créez un fichier nommé `run_script.rb` avec le code suivant. Pour plus d'informations, consultez 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. Depuis le terminal ou l'invite de commande, utilisez la commande tar pour créer une nouvelle version du fichier `opsworks_cookbook_demo.tar.gz`, qui contient le répertoire `opsworks_cookbook_demo` et son contenu chargé.

3. Chargez le fichier `opsworks_cookbook_demo.tar.gz` mis à jour dans votre compartiment S3.
4. Suivez les procédures décrites dans Étape 5 : Mettre à jour le livre de recettes sur l'instance et exécuter la recette pour mettre à jour le livre de recettes sur l'instance et exécuter la recette. Dans la procédure « Pour exécuter la recette », pour Recipes to execute (Recettes à exécuter), tapez **`opsworks_cookbook_demo::run_script`**.

Pour tester la recette

1. Connectez-vous à l'instance, si vous le n'avez pas déjà fait.
2. Depuis l'invite de commande, exécutez la commande suivante afin de confirmer que le nouveau fichier a été ajouté :

   ```
   sudo cat /tmp/run-script-demo/helloworld.txt
   ```

   Le contenu du fichier s'affiche :

   ```
   Hello, World!
   ```

Dans l'étape suivante, vous allez mettre à jour le livre de recettes pour gérer un service sur l'instance.

Étape 11 : Mettre à jour le livre de recettes pour gérer un service

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Mettez à jour votre livre de recettes en ajoutant une recette qui gère un service sur l'instance. Cette action est similaire à l'exécution de la commande Linux service ou des commandes Windows net stop, net start et autres commandes similaires. Cette recette arrête le service crond sur l'instance.

Pour mettre à jour le livre de recettes sur l'instance et exécuter la nouvelle recette

1. Sur votre ordinateur local, dans le sous-répertoire `recipes` du répertoire `opsworks_cookbook_demo`, créez un fichier nommé `manage_service.rb` avec le code suivant. Pour plus d'informations, consultez service.

   ```
   service "Manage a service" do
     action :stop
     service_name "crond"
   end
   ```

2. Depuis le terminal ou l'invite de commande, utilisez la commande tar pour créer une nouvelle version du fichier `opsworks_cookbook_demo.tar.gz`, qui contient le répertoire `opsworks_cookbook_demo` et son contenu chargé.
3. Chargez le fichier `opsworks_cookbook_demo.tar.gz` mis à jour dans votre compartiment S3.
4. Suivez les procédures décrites dans Étape 5 : Mettre à jour le livre de recettes sur l'instance et exécuter la recette pour mettre à jour le livre de recettes sur l'instance et exécuter la recette. Dans la procédure « Pour exécuter la recette », pour Recipes to execute (Recettes à exécuter), tapez **`opsworks_cookbook_demo::manage_service`**.

Pour tester la recette

1. Connectez-vous à l'instance, si vous le n'avez pas déjà fait.
2. À l'invite de commande, exécutez la commande suivante afin de confirmer que le service crond est arrêté :

   ```
   service crond status
   ```

   Les informations suivantes s'affichent :

   ```
   crond is stopped
   ```

3. Pour redémarrer le service crond, exécutez la commande suivante :

   ```
   sudo service crond start
   ```

   Les informations suivantes s'affichent :

```
Starting crond:  [  OK  ]
```

4. Pour confirmer que le service crond a démarré, exécutez à nouveau la commande suivante :

   ```
   service crond status
   ```

   Les informations telles que les suivantes s'affichent :

   ```
   crond (pid  3917) is running...
   ```

Dans l'étape suivante, vous allez mettre à jour le livre de recettes pour faire référence aux informations stockées en tant que JSON personnalisé sur l'instance.

Étape 12 : Mettre à jour le livre de recettes pour utiliser JSON personnalisé

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Mettez à jour votre livre de recettes en ajoutant une recette qui fait référence au JSON personnalisé stocké sur l'instance.

Vous pouvez spécifier des informations au format JSON personnalisé chaque fois que vous créez, mettre à jour ou clonez une pile, ou lorsque vous exécutez une commande de pile ou de déploiement. Cela est utile, par exemple, pour rendre une petite portion invariable de données accessible à vos recettes sur l'instance au lieu d'obtenir ces données à partir d'une base de données. Pour de plus amples informations, veuillez consulter Utilisation du JSON personnalisé.

Pour cette procédure pas à pas, vous allez utiliser JSON personnalisé pour fournir des informations fictives à propos d'une facture client. Le JSON personnalisé est décrit ultérieurement dans cette étape.

Pour mettre à jour le livre de recettes sur l'instance et exécuter la nouvelle recette

1. Sur votre ordinateur local, dans le sous-répertoire `recipes` du répertoire `opsworks_cookbook_demo`, créez un fichier nommé `custom_json.rb` qui contient le code de recette suivant :

   ```
   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']}' **********")
   ```

   Cette recette affiche des messages dans le journal relatifs aux valeurs du JSON personnalisé.

2. Depuis le terminal ou l'invite de commande, utilisez la commande tar pour créer une nouvelle version du fichier `opsworks_cookbook_demo.tar.gz`, qui contient le répertoire `opsworks_cookbook_demo` et son contenu chargé.

3. Chargez le fichier `opsworks_cookbook_demo.tar.gz` mis à jour dans votre compartiment S3.

4. Suivez les procédures décrites dans Étape 5 : Mettre à jour le livre de recettes sur l'instance et exécuter la recette pour mettre à jour le livre de recettes sur l'instance et exécuter la recette. Dans la procédure « Pour exécuter la recette », pour Recipes to execute (Recettes à exécuter), tapez **`opsworks_cookbook_demo::custom_json`**. Pour Advanced (Avancé), Custom Chef JSON (JSON Chef personnalisé), tapez le JSON personnalisé suivant :

   ```
   {
     "customer-id": "0123",
     "invoice-number": "9876",
     "line-items": {
       "line-1": "tractor",
       "line-2": "passenger car",
       "line-3": "trailer"
     }
   }
   ```

Pour tester la recette

1. Avec la page Running command execute_recipes (Exécution de la commande execute_recipes) affichée à partir des procédures précédentes, pour cookbooks-demo1, pour Log (Journal), choisissez show (afficher). La page de journal execute_recipes s'affiche.
2. Faites défiler le journal pour trouver les entrées similaires aux entrées suivantes :

```
[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'
 **********
```

   Ces entrées affichent les informations du JSON personnalisé saisies dans la zone Advanced (Avancé), Custom Chef JSON (JSON Chef personnalisé).

À l'étape suivante, vous allez mettre à jour le livre de recettes pour obtenir des informations provenant des sacs de données, qui sont des ensembles de paramètres de pile que AWS OpsWorks Stacks stocke sur chaque instance.

Étape 13 : Mettre à jour le livre de recettes pour utiliser les conteneurs de données

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Mettez à jour votre livre de recettes en ajoutant une recette qui fait référence aux paramètres de pile que AWS OpsWorks Stacks stocke sur l'instance dans un ensemble de sacs de données. Cette recette affiche des messages dans le journal relatifs aux paramètres de pile spécifiques stockés sur

l'instance. Pour de plus amples informations, veuillez consulter AWS OpsWorks Référence du sac de données Stacks.

Pour mettre à jour le livre de recettes sur l'instance et exécuter la nouvelle recette

1. Sur votre ordinateur local, dans le sous-répertoire `recipes` du répertoire `opsworks_cookbook_demo`, créez un fichier nommé `data_bags.rb` qui contient le code suivant :

```
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']}' **********")
```

   Cette recette affiche des messages dans le journal relatifs aux paramètres de pile spécifiques stockés sur l'instance.

2. Depuis le terminal ou l'invite de commande, utilisez la commande tar pour créer une nouvelle version du fichier `opsworks_cookbook_demo.tar.gz`, qui contient le répertoire `opsworks_cookbook_demo` et son contenu chargé.
3. Chargez le fichier `opsworks_cookbook_demo.tar.gz` mis à jour dans votre compartiment S3.
4. Suivez les procédures décrites dans Étape 5 : Mettre à jour le livre de recettes sur l'instance et exécuter la recette pour mettre à jour le livre de recettes sur l'instance et exécuter la recette. Dans la procédure « Pour exécuter la recette », pour Recipes to execute (Recettes à exécuter), tapez **`opsworks_cookbook_demo::data_bags`**.

Pour tester la recette

1. Avec la page Running command execute_recipes (Exécution de la commande execute_recipes) affichée à partir de la procédure précédente, pour cookbooks-demo1, pour Log (Journal), choisissez show (afficher). La page de journal execute_recipes s'affiche.
2. Faites défiler le journal pour trouver les entrées similaires aux entrées suivantes :

```
[2015-11-14T14:39:06+00:00] INFO: ********** This instance's instance ID is
 'f80fa119-81ab-4c3c-883d-6028e52c89EX' **********
[2015-11-14T14:39:06+00:00] INFO: ********** This instance's public IP address is
 '192.0.2.0' **********
[2015-11-14T14:39:06+00:00] INFO: ********** This instance belongs to the layer
 'MyCookbooksDemoLayer' **********
[2015-11-14T14:39:06+00:00] INFO: ********** This instance belongs to the stack
 'MyCookbooksDemoStack' **********
[2015-11-14T14:39:06+00:00] INFO: ********** This stack gets its cookbooks from
 'https://s3.amazonaws.com/amzn-s3-demo-bucket/opsworks_cookbook_demo.tar.gz'
 **********
```

   Cette recette affiche des messages relatifs aux paramètres de pile spécifiques stockés sur l'instance.

Dans l'étape suivante, vous allez mettre à jour le livre de recettes pour exécuter le code de la recette plusieurs fois.

Étape 14 : Mettre à jour le livre de recettes pour utiliser l'itération

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Mettez à jour votre livre de recettes en ajoutant une recette qui utilise l'itération, technique qui répète le code de la recette plusieurs fois. Cette recette affiche les messages du journal relatifs à un élément de conteneur de données qui inclut un contenu multiple.

Pour mettre à jour le livre de recettes sur l'instance et exécuter la nouvelle recette

1. Sur votre ordinateur local, dans le sous-répertoire `recipes` du répertoire `opsworks_cookbook_demo`, créez un fichier nommé `iteration_demo.rb` qui contient le code suivant :

```
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**
>
> L'écriture du code de recette précédent est plus courte, plus souple et plus fiable que l'écriture du code suivant qui n'utilise pas l'itération :
>
> ```
> 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. Depuis le terminal ou l'invite de commande, utilisez la commande tar pour créer une nouvelle version du fichier `opsworks_cookbook_demo.tar.gz`, qui contient le répertoire `opsworks_cookbook_demo` et son contenu chargé.
3. Chargez le fichier `opsworks_cookbook_demo.tar.gz` mis à jour dans votre compartiment S3.

4. Suivez les procédures décrites dans Étape 5 : Mettre à jour le livre de recettes sur l'instance et exécuter la recette pour mettre à jour le livre de recettes sur l'instance et exécuter la recette. Dans la procédure « Pour exécuter la recette », pour Recipes to execute (Recettes à exécuter), tapez **`opsworks_cookbook_demo::iteration_demo`**.

Pour tester la recette

1. Avec la page Running command execute_recipes (Exécution de la commande execute_recipes) affichée à partir des procédures précédentes, pour cookbooks-demo1, pour Log (Journal), choisissez show (afficher). La page de journal execute_recipes s'affiche.
2. Faites défiler le journal pour trouver les entrées similaires aux entrées suivantes :

```
[2015-11-16T19:56:56+00:00] INFO: ********** Content of 'custom_cookbooks_source'
 **********
[2015-11-16T19:56:56+00:00] INFO: ********** '["type", "s3"]' **********
[2015-11-16T19:56:56+00:00] INFO: ********** '["url", "https://s3.amazonaws.com/
amzn-s3-demo-bucket/opsworks_cookbook_demo.tar.gz"]' **********
[2015-11-16T19:56:56+00:00] INFO: ********** '["username", "secret-key-value"]'
 **********
[2015-11-16T19:56:56+00:00] INFO: ********** '["password", "secret-access-key-
value"]' **********
[2015-11-16T19:56:56+00:00] INFO: ********** '["ssh_key", nil]' **********
[2015-11-16T19:56:56+00:00] INFO: ********** '["revision", nil]' **********
```

   Cette recette affiche les messages du journal relatifs à un élément de conteneur de données qui inclut un contenu multiple. L'élément de conteneur de données est dans le conteneur de données `aws_opsworks_stack`. L'élément de conteneur de données possède un contenu nommé `custom_cookbooks_source`. A l'intérieur de ce contenu figurent six contenus nommés `type`, `url`, `username`, `password`, `ssh_key` et `revision` ; leurs valeurs sont également affichées.

Dans l'étape suivante, vous allez mettre à jour le livre de recettes pour n'exécuter le code de la recette que dans certaines conditions.

Étape 15 : Mettre à jour le livre de recettes pour utiliser la logique conditionnelle

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Mettez maintenant à jour votre livre de recettes en ajoutant une recette qui utilise la logique conditionnelle, technique qui n'exécute le code que si certaines conditions sont réunies. Pour plus d'informations, consultez Instructions if et Instructions case.

Cette recette se compose de deux actions basées sur le contenu des conteneurs de données : affiche un message dans le journal identifiant le système d'exploitation sur lequel l'instance s'exécute et, uniquement si le système d'exploitation est Linux, installe un package en utilisant le gestionnaire de package approprié pour la distribution Linux donnée. Ce package est appelé tree (arborescence) ; il s'agit d'une simple application pour visualiser les listes de répertoire.

Pour mettre à jour le livre de recettes sur l'instance et exécuter la nouvelle recette

1. Sur votre ordinateur local, dans le sous-répertoire `recipes` de `opsworks_cookbook_demo directory`, créez un fichier nommé `conditional_logic.rb` qui contient le code suivant :

```
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. Depuis le terminal ou l'invite de commande, utilisez la commande tar pour créer une nouvelle version du fichier `opsworks_cookbook_demo.tar.gz`, qui contient le répertoire `opsworks_cookbook_demo` et son contenu chargé.
3. Chargez le fichier `opsworks_cookbook_demo.tar.gz` mis à jour dans votre compartiment S3.
4. Suivez les procédures décrites dans Étape 5 : Mettre à jour le livre de recettes sur l'instance et exécuter la recette pour mettre à jour le livre de recettes sur l'instance et exécuter la recette. Dans la procédure « Pour exécuter la recette », pour Recipes to execute (Recettes à exécuter), tapez **`opsworks_cookbook_demo::conditional_logic`**.

Pour tester la recette

1. Avec la page Running command execute_recipes (Exécution de la commande execute_recipes) affichée à partir des procédures précédentes, pour cookbooks-demo1, pour Log (Journal), choisissez show (afficher). La page de journal execute_recipes s'affiche.
2. Faites défiler le journal et recherchez une entrée qui ressemble à ce qui suit :

```
[2015-11-16T19:59:05+00:00] INFO: ********** Operating system is Amazon Linux.
 **********
```

Le système d'exploitation de l'instance étant Amazon Linux 2016.09, seule l'entrée précédente (des cinq entrées possibles dans le code de la recette) sera affichée dans le journal.

3. Si le système d'exploitation est Linux, la recette installe le package tree. Pour visualiser le contenu d'un répertoire, tapez **`tree`** à l'invite de commande depuis le répertoire souhaité ou avec le chemin d'accès du répertoire souhaité (par exemple, `tree /var/chef/runs`).

Dans l'étape suivante, vous allez mettre à jour le livre de recettes pour utiliser les fonctionnalités à partir d'un livre de recettes externe fourni par la communauté Chef.

Étape 16 : Mettre à jour le livre de recettes pour utiliser les livres de recettes de la communauté

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Enfin, mettez à jour le livre de recettes pour utiliser les fonctionnalités fournies dans un livre de recettes externe fourni par la communauté Chef. Le livre de recettes externe que vous utilisez pour cette procédure pas à pas est disponible via le supermarché Chef, célèbre endroit pour accéder aux livres de recettes Chef externes. Ce livre de recettes externe fournit une ressource personnalisée qui vous permet de télécharger et d'installer des applications, de la même manière que ce que vous avez fait dans Étape 4 : Mettre à jour le livre de recettes pour installer un package. Cependant, cette ressource peut installer des applications web et autres types d'applications en plus des packages.

Quand un livre de recettes repose sur un autre livre de recettes, vous devez spécifier une dépendance sur les autres livres de recettes. Pour déclarer et gérer les dépendances de livre de recettes, nous recommandons d'utiliser un outil appelé Berkshelf. Pour plus d'informations sur la

façon d'installer Berkshelf sur votre poste de travail local, consultez About Berkshelf (À propos de Berkshelf) sur le site web de Chef.

Une fois que vous avez installé Berkshelf, suivez ces procédures pour déclarer la dépendance du livre de recettes, puis créez une recette qui appelle la ressource du livre de recettes externe :

Pour déclarer la dépendance du livre de recettes

1. Sur votre ordinateur local, dans le répertoire `opsworks_cookbook_demo`, ajoutez la ligne suivante à la fin du fichier `metadata.rb` :

   ```
   depends "application", "5.0.0"
   ```

   Cette action déclare une dépendance sur un livre de recettes intitulé `application`, version 5.0.0.

2. Depuis la racine du répertoire `opsworks_cookbook_demo`, exécutez la commande suivante. Le point à la fin de la commande est intentionnel.

   ```
   berks init .
   ```

   Berkshelf crée un certain nombre de dossiers et de fichiers que vous pouvez utiliser ultérieurement pour les scénarios plus avancés. Le seul fichier dont nous avons besoin pour cette procédure pas à pas est le fichier nommé `Berksfile`.

3. Ajoutez la ligne suivante à la fin du fichier `Berksfile` :

   ```
   cookbook "application", "5.0.0"
   ```

   Cette action informe Berkshelf que vous voulez utiliser la version 5.0.0 du livre de recettes « application », que Berkshelf télécharge via le Chef Supermarket.

4. Depuis un terminal ou une invite de commande, exécutez la commande suivante à partir de la racine du répertoire `opsworks_cookbook_demo` :

   ```
   berks install
   ```

   Berkshelf crée une liste de dépendances pour votre livre de recettes et pour le livre de recettes « application ». Berkshelf utilise cette liste de dépendances dans la procédure suivante.

Pour mettre à jour le livre de recettes sur l'instance et exécuter la nouvelle recette

1. Dans le sous-répertoire `recipes` du répertoire `opsworks_cookbook_demo`, créez un fichier nommé `dependencies_demo.rb` qui contient le code suivant :

   ```
   application "Install NetHack" do
     package "nethack.x86_64"
   end
   ```

   Cette recette dépend de la ressource d'application contenue dans le livre de recettes de l'application pour installer le célèbre jeu d'aventure textuel NetHack sur l'instance. (Vous pouvez, bien sûr, utiliser le nom de package de votre choix, à condition que le package soit facilement accessible pour le gestionnaire de package sur l'instance.)

2. Depuis la racine du répertoire `opsworks_cookbook_demo`, exécutez la commande suivante :

   ```
   berks package
   ```

   Berkshelf utilise la liste des dépendances de la procédure précédente pour créer un fichier nommé `cookbooks-`*`timestamp`*`.tar.gz`, qui contient le répertoire `opsworks_cookbook_demo` et son contenu mis à jour, y compris les livres de recettes dépendants du livre de recettes. Renommez le fichier `opsworks_cookbook_demo.tar.gz`.

3. Chargez le fichier renommé `opsworks_cookbook_demo.tar.gz` mis à jour dans votre compartiment S3.

4. Suivez les procédures décrites dans Étape 5 : Mettre à jour le livre de recettes sur l'instance et exécuter la recette pour mettre à jour le livre de recettes sur l'instance et exécuter la recette. Dans la procédure « Pour exécuter la recette », pour Recipes to execute (Recettes à exécuter), tapez **`opsworks_cookbook_demo::dependencies_demo`**.

5. Une fois que vous avez exécuté la recette, vous devriez pouvoir vous connecter à l'instance, puis taper **`nethack`** à l'invite de commande pour commencer à jouer. (Pour plus d'informations sur le jeu, consultez NetHacket le NetHackGuide.)

À l'étape suivante, vous pouvez nettoyer les AWS ressources que vous avez utilisées pour cette procédure pas à pas. Cette étape suivante est facultative. Vous souhaiterez peut-être continuer à utiliser ces AWS ressources tout en continuant à en apprendre davantage sur AWS OpsWorks Stacks. Cependant, la conservation de ces AWS ressources peut entraîner des frais permanents sur

votre AWS compte. Si vous souhaitez conserver ces AWS ressources pour une utilisation ultérieure, vous avez maintenant terminé cette procédure pas à pas et vous pouvez passer àÉtapes suivantes.

Étape 17 : (Facultatif) Nettoyer

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Pour éviter d'encourir des frais supplémentaires sur votre AWS compte, vous pouvez supprimer les AWS ressources utilisées pour cette procédure pas à pas. Ces AWS ressources incluent le compartiment S3, la pile AWS OpsWorks Stacks et les composants de la pile. (Pour de plus amples informations, veuillez consulter OpsWorks Tarification AWS.) Cependant, vous souhaiterez peut-être continuer à utiliser ces AWS ressources tout en continuant à en apprendre davantage sur AWS OpsWorks Stacks. Si vous souhaitez que ces AWS ressources restent disponibles, vous avez maintenant terminé cette procédure pas à pas et vous pouvez passer àÉtapes suivantes.

Le contenu stocké dans les ressources que vous avez créées pour cette procédure pas à pas peut contenir des informations d'identification personnelle. Si vous ne voulez plus que ces informations soient stockées par AWS, suivez les étapes décrites dans cette rubrique.

Pour supprimer le compartiment S3

- Consultez Supprimer le compartiment Amazon S3.

Pour supprimer l'instance de la pile

1. Dans la console AWS OpsWorks Stacks, dans le volet de navigation des services, choisissez Instances (Instances). La page Instances s'affiche.
2. Pour MyCookbooksDemoLayer, pour cookbooks-demo1, pour Actions, choisissez stop. Dans le message de confirmation, sélectionnez Stop.
3. Les modifications suivantes prennent plusieurs minutes. Ne continuez pas tant que toutes les opérations suivantes ne sont pas terminées.

- Status (Statut) passe de online (en ligne) à stopping (en cours d'arrêt) et, enfin, à stopped (arrêté).
- online (en ligne) passe de 1 à 0.
- shutting down (en cours de fermeture) passe de 0 à 1 et, enfin, à 0.
- stopped (arrêté) passe finalement de 0 à 1.

4. Pour Actions, choisissez delete (supprimer). Lorsque le message de confirmation s'affiche, choisissez Supprimer. AWS OpsWorks Stacks supprime l'instance et affiche Aucune instance.

Pour supprimer la pile

1. Dans le panneau de navigation du service, choisissez Stack (Pile). La page MyCookbooksDemoStack s'affiche.
2. Choisissez Supprimer pile. Lorsque le message de confirmation s'affiche, choisissez Supprimer. AWS OpsWorks Stacks supprime la pile et affiche la page du tableau de bord.

Vous pouvez éventuellement supprimer l'utilisateur IAM et la paire de EC2 clés Amazon que vous avez utilisés pour cette procédure pas à pas, si vous ne souhaitez pas les réutiliser pour accéder à d'autres AWS services et EC2 instances. Pour obtenir des instructions, consultez Supprimer un utilisateur IAM, des paires de EC2 clés Amazon et des instances Linux.

Vous avez maintenant terminé cette procédure pas à pas. Pour de plus amples informations, veuillez consulter Étapes suivantes.

Étapes suivantes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Maintenant que vous avez terminé cette procédure pas à pas, vous pouvez en savoir plus sur le support de AWS OpsWorks Stacks pour les livres de cuisine Chef en consultant les ressources suivantes :

- Livres de recettes et recettes— Décrit les versions de Chef et Ruby actuellement AWS OpsWorks prises en charge par Stacks. Explique aussi comment installer et mettre à jour les livres de recettes personnalisés sur les instances et comment exécuter des recettes sur les instances.
- Learn Chef — Fournit des liens vers des didacticiels Chef, une bibliothèque de compétences Chef, une documentation complète sur Chef et des cours de formation Chef.
- Tout sur Chef — Fournit une documentation complète sur Chef. Parmi les thèmes d'intérêt spécifiques, citons :
  - À propos des livres de recettes — Décrit les principaux composants des livres de recettes tels que les attributs, les recettes, les fichiers, les métadonnées et les modèles.
  - À propos des recettes — Décrit les principes fondamentaux des recettes, tels que la façon d'utiliser les sacs de données, d'inclure d'autres recettes et d'utiliser le code Ruby dans les recettes.
  - Ressources — Décrit comment utiliser toutes les ressources intégrées de Chef, telles que `apt_package`,`cookbook_file`,`directory`,`execute`,`file`, et`package`.
  - À propos du Recipe DSL — Décrit comment écrire du code pour les recettes Chef avec des instructions telles que `if`,`case`,`data_bag`,`data_bag_item`, et`search`.
- À propos des modèles — Décrit comment utiliser les modèles Embedded Ruby (ERB) pour générer dynamiquement des fichiers texte statiques, tels que des fichiers de configuration.
- Pistes de formation : décrit comment utiliser Chef pour gérer une instance, gérer une application Web de base, développer et tester du code d'infrastructure, utiliser les analyses Chef, etc.
- Learning Chef — Une introduction à Chef. Publié par O'Reilly Media.
- Exemples de code Learning Chef — Fournit des exemples de code pour accompagner le livre Learning Chef publié par O'Reilly Media.

AWS OpsWorks Meilleures pratiques de Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients

de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les stratégies, techniques et suggestions présentées dans cette section peuvent vous aider à tirer le meilleur parti de AWS OpsWorks Stacks et à obtenir des résultats optimaux.

Rubriques

- Bonnes pratiques : stockage de périphérique racine pour les instances
- Bonnes pratiques : optimisation du nombre de serveurs d'applications
- Bonnes pratiques : Gestion des autorisations
- Bonnes pratiques : Gestion et déploiement des applications et des livres de recettes
- Empaquetage local des dépendances des livres de recettes

Bonnes pratiques : stockage de périphérique racine pour les instances

Important

Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Note

Cette rubrique ne s'applique pas aux instances Windows, qui doivent être soutenues par Amazon Elastic Block Store.

Les instances Linux Amazon Elastic Compute Cloud (Amazon EC2) disposent des options de stockage sur périphérique racine suivantes.

- Instances basées sur le stockage d'instances : le périphérique racine est temporaire.

Si vous arrêtez l'instance, les données du périphérique racine disparaissent et ne peuvent pas être récupérées. Pour plus d'informations, consultez Amazon EC2 Instance Store.

- Instances basées sur Amazon EBS : le périphérique racine est un volume Amazon EBS.

  Si vous arrêtez l'instance, le volume Amazon EBS persiste. Si vous redémarrez l'instance, le volume est automatiquement remonté, ce qui restaure l'état de l'instance et les données stockées. Vous pouvez également monter le volume sur une autre instance. Pour plus d'informations, consultez Amazon Elastic Block Store (Amazon EBS).

Prenez en compte les éléments suivants au moment de décider quelle option de stockage de périphérique racine vous allez utiliser.

Durée du démarrage

Après le démarrage initial, les instances Amazon EBS redémarrent généralement plus rapidement.

Le temps de démarrage initial est environ le même pour chaque type de stockage. Les deux types doivent effectuer une installation complète, ce qui inclut des tâches relativement gourmandes en temps, comme l'installation de packages à partir de référentiels distants. Cependant, prenez en compte ces différences lorsque vous redémarrez par la suite une instance :

- Les instances basées sur le stockage d'instance effectuent les mêmes tâches d'installation que celles exécutées pour le démarrage initial, y compris l'installation des packages.

  Un redémarrage nécessite à peu près le même temps que le démarrage initial.

- Les instances Amazon EBS-Back remontent le volume racine et exécutent les recettes de configuration.

  Le redémarrage est généralement beaucoup plus rapide que le démarrage initial, car les recettes Setup n'ont pas à effectuer des tâches telles que la réinstallation de packages qui sont déjà installés sur le volume racine.

Coût

Les instances basées sur Amazon EBS sont plus coûteuses :

- Avec une instance basée sur le stockage d'instance, vous payez uniquement lorsque l'instance est en cours d'exécution.

- Avec les instances basées sur Amazon EBS, vous payez pour le volume Amazon EBS, que l'instance soit en cours d'exécution ou non.

  Pour plus d'informations, consultez Tarification Amazon EBS.

Journalisation

Les instances basées sur Amazon EBS conservent automatiquement les journaux :

- Avec l'instance basée sur le stockage d'instance, les journaux disparaissent lorsque l'instance s'arrête.

  Vous devez soit récupérer les journaux avant d'arrêter l'instance, soit utiliser un service tel que CloudWatch Logs pour stocker à distance les journaux sélectionnés.
- Avec une instance basée sur Amazon EBS, les journaux sont stockés sur le volume Amazon EBS.

  Vous pouvez les afficher en redémarrant l'instance ou en montant le volume sur une autre instance.

Dépendances

Les deux types de stockage ont des dépendances différentes :

- Les instances soutenues par un magasin d'instances dépendent d'Amazon S3.

  Lorsque vous démarrez l'instance, elle doit télécharger l'AMI depuis Amazon S3.
- Les instances soutenues par Amazon EBS dépendent d'Amazon EBS.

  Lorsque vous démarrez l'instance, elle doit monter le volume racine Amazon EBS.

Recommandation : si vous n'êtes pas certain du type de stockage le mieux adapté à vos besoins, nous vous recommandons de commencer par les instances Amazon EBS. Bien que les volumes Amazon EBS entraînent des dépenses modestes, le risque de perte de données involontaire est moindre.

Bonnes pratiques : optimisation du nombre de serveurs d'applications

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients

de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une pile en production comprend généralement plusieurs serveurs d'applications répartis entre plusieurs zones de disponibilité. Cependant, le nombre de demandes entrantes peut varier considérablement en fonction de l'heure ou du jour de la semaine. Vous pouvez juste exécuter assez de serveurs pour gérer la charge maximale prévue, mais la plupart du temps vous finirez par payer une plus grande capacité serveur que celle dont vous avez besoin. Pour gérer votre site de manière efficace, la pratique recommandée consiste à faire correspondre le nombre de serveurs et le volume de demandes en cours.

AWS OpsWorks Stacks propose trois méthodes pour gérer le nombre d'instances de serveur.

- Les instances 24 h/24 et 7 j/7 sont démarrées manuellement et exécutées jusqu'à ce qu'elles soient arrêtées manuellement.
- Les instances basées sur le temps sont automatiquement démarrées et arrêtées par AWS OpsWorks Stacks selon un calendrier défini par l'utilisateur.
- Les instances basées sur la charge sont automatiquement démarrées et arrêtées par AWS OpsWorks Stacks lorsqu'elles franchissent un seuil pour une métrique de charge spécifiée par l'utilisateur, telle que l'utilisation du processeur ou de la mémoire.

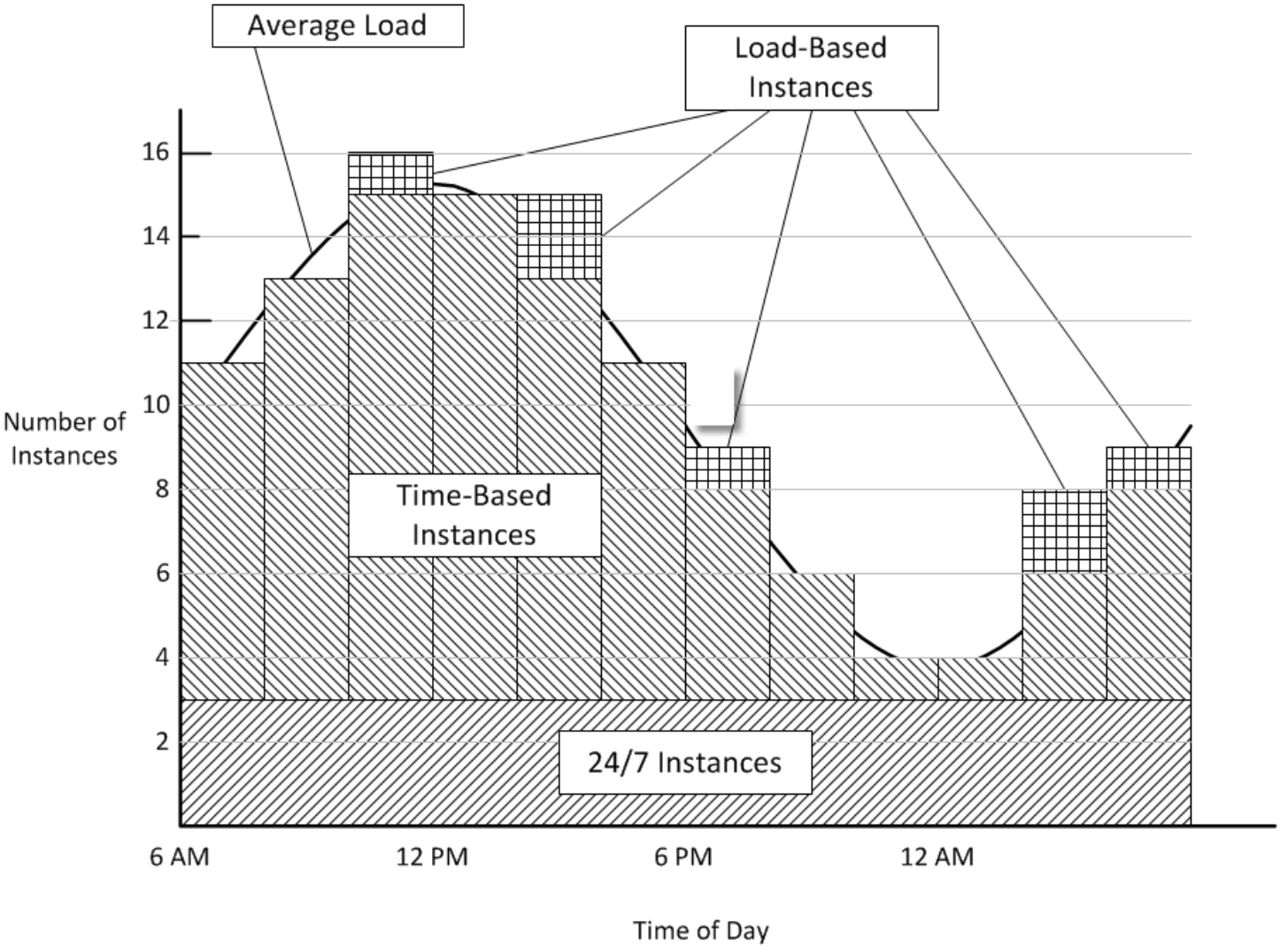
> **Note**
>
> Une fois que vous avez créé et configuré les instances basées sur la date et les instances basées sur la charge de votre pile, AWS OpsWorks Stacks les démarre et les arrête automatiquement en fonction de la configuration spécifiée. Vous n'avez pas à les retoucher, sauf si vous décidez de changer la configuration ou le nombre d'instances.

Recommandation : si vous gérez les piles avec plusieurs instances de serveur d'applications, nous recommandons d'utiliser un mélange des trois types d'instances. Voici un exemple de la façon de gérer la capacité serveur d'une pile pour traiter un volume variable de demandes quotidiennes aux caractéristiques suivantes.

- Le volume moyen des demandes varie de façon sinusoïdale au cours de la journée.

- Le volume moyen minimal de demandes nécessite cinq instances serveur d'applications.
- Le volume moyen maximal de demandes nécessite seize instances serveur d'applications.
- Les pics du volume de demandes peuvent généralement être traités par une ou deux instances serveur d'applications.

Il s'agit d'un modèle pratique dans le cadre de cette discussion, mais vous pouvez facilement l'adapter à toute modification du volume des demandes et l'étendre également pour gérer les variations hebdomadaires. Le schéma suivant montre comment utiliser les trois types d'instance pour gérer ce volume de demandes.



Cet exemple possède les caractéristiques suivantes :

- La pile possède trois instances 24/7, qui sont toujours actives et gèrent la charge de base.

- La pile a 12 instances basées sur la date, qui sont configurées pour gérer la variation quotidienne moyenne.

  L'une s'exécute de 10 pm à 2 am, deux autres s'exécutent de 8 pm à 10 pm et de 2 am à 4 am, etc. Pour plus de simplicité, le schéma modifie toutes les deux heures le nombre d'instances basées sur la date, mais vous pouvez modifier le nombre toutes les heures si vous souhaitez un contrôle plus fin.
- La pile a suffisamment d'instances basées sur la charge pour gérer les pics de trafic qui dépassent ce qui peut être géré par les instances 24/7 et les instances basées sur l'heure.

  AWS OpsWorks Stacks démarre des instances basées sur la charge uniquement lorsque la charge sur tous les serveurs en cours d'exécution dépasse les métriques spécifiées. Le coût des instances non opérationnelles est minime (instances basées sur Amazon EBS) ou nul (instances sauvegardées par le stockage d'instances). Il est donc recommandé d'en créer suffisamment pour gérer confortablement le volume maximal de demandes prévu. Pour cet exemple, la pile doit avoir au moins trois instances basées sur la charge.

> **Note**
>
> Assurez-vous d'avoir les trois types d'instance répartis entre plusieurs zones de disponibilité pour réduire l'impact des perturbations de tel ou tel service.

Bonnes pratiques : Gestion des autorisations

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous devez disposer d'informations d'identification AWS pour accéder aux ressources de votre compte. Voici quelques conseils généraux pour donner accès à vos employés.

- Tout d'abord, nous vous déconseillons d'utiliser les informations d'identification racine de votre compte pour accéder aux ressources AWS.

  Créez plutôt des identités IAM pour vos employés et ajoutez des autorisations qui fournissent un accès approprié. Chaque employé peut ensuite utiliser ses informations d'identification pour accéder aux ressources.
- Les employés doivent disposer d'autorisations pour accéder uniquement aux ressources dont ils ont besoin pour effectuer leurs tâches.

  Par exemple, les développeurs d'applications ont besoin d'accéder uniquement aux piles qui exécutent leurs applications.
- Les employés doivent disposer d'autorisations pour utiliser uniquement les actions nécessaires à la réalisation de leurs tâches.

  Un développeur d'applications peut avoir besoin d'autorisations complètes pour une pile de développement et d'autorisations pour déployer ses applications sur la pile de production correspondante. Ils n'ont probablement pas besoin des autorisations permettant de démarrer ou d'arrêter les instances sur la pile de production, de créer ou de supprimer des couches, etc.

Pour plus d'informations générales sur la gestion des autorisations, consultez Informations d'identification de sécurité AWS.

Vous pouvez utiliser AWS OpsWorks Stacks ou IAM pour gérer les autorisations des utilisateurs. Notez que les deux options ne s'excluent pas mutuellement. Il convient parfois d'utiliser les deux.

AWS OpsWorks Gestion des autorisations de Stacks

Chaque pile a une page Permissions (Autorisations) que vous pouvez utiliser pour donner aux utilisateurs les autorisations d'accès à la pile et spécifier les actions qu'ils peuvent effectuer. Vous spécifiez les autorisations d'un utilisateur en définissant l'un des niveaux d'autorisation suivants. Chaque niveau représente une politique IAM qui accorde des autorisations pour un ensemble d'actions standard.

- Deny (Refuser) refuse l'autorisation d'interagir avec la pile de quelque façon que ce soit.
- Show (Afficher) octroie des autorisations d'affichage de la configuration de la pile, mais pas de modification de l'état de la pile de quelque façon que ce soit.
- Deploy (Déployer) inclut les autorisations Show (Afficher) et octroie également des autorisations de déploiement des applications aux utilisateurs.

- Manage (Gérer) inclut les autorisations Deploy (Déployer) et permet également à l'utilisateur d'effectuer diverses actions de gestion de la pile, telles que la création ou la suppression des couches et des instances.

> **Note**
>
> Le niveau Gérer les autorisations n'accorde pas d'autorisations pour un petit nombre d'actions de haut niveau dans les AWS OpsWorks Stacks, y compris la création ou le clonage de piles. Vous devez utiliser une politique IAM pour accorder ces autorisations.

Outre la définition des niveaux d'autorisation, vous pouvez également utiliser la page Permissions d'une pile pour indiquer si les utilisateurs ont des SSH/RDP and sudo/admin privilèges sur les instances de la pile. Pour plus d'informations sur la gestion des autorisations AWS OpsWorks Stacks, consultez Attribution des autorisations par pile. Pour plus d'informations sur la gestion de l'accès SSH, consultez Gestion de l'accès SSH.

Gestion des autorisations IAM

Avec la gestion des autorisations IAM, vous utilisez la console, l'API ou la CLI IAM pour associer une politique au format JSON à un utilisateur qui spécifie explicitement ses autorisations. Pour plus d'informations sur la gestion des autorisations IAM, voir Qu'est-ce que l'IAM ? .

Recommandation : Commencez par la gestion des autorisations AWS OpsWorks Stacks. Si vous avez besoin d'affiner les autorisations d'un utilisateur ou d'accorder à un utilisateur des autorisations qui ne sont pas incluses dans les niveaux d'autorisation Manage (Gérer), vous pouvez combiner les deux approches. AWS OpsWorks Stacks évalue ensuite les deux politiques pour déterminer les autorisations de l'utilisateur.

> **Important**
>
> Si un utilisateur applique plusieurs politiques avec des autorisations contradictoires, le refus l'emporte toujours. Supposons, par exemple, que vous associez une politique IAM à un utilisateur qui autorise l'accès à une pile particulière, mais que vous utilisiez également la page Autorisations de la pile pour attribuer à l'utilisateur un niveau d'autorisation de refus. Les autorisations Deny (Refuser) sont prioritaires et l'utilisateur ne sera pas en mesure d'accéder à la pile. Pour plus d'informations, consultez la section Logique d'évaluation des politiques IAM.

Par exemple, supposons que vous souhaitiez qu'un utilisateur puisse effectuer la plupart des opérations sur une pile, à l'exception de l'ajout ou de la suppression des couches.

- Spécifiez un niveau d'autorisations Manage (Gérer) qui permet à l'utilisateur d'effectuer la plupart des actions de gestion de la pile, y compris la création et la suppression des couches.
- Attachez à l'utilisateur la politique gérée par le client suivante, qui refuse l'autorisation d'utiliser les DeleteLayeractions CreateLayeret sur cette pile. Vous identifiez la pile grâce à son nom de ressource Amazon (ARN), qui se trouve sur la page Settings (Paramètres) de la pile.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "opsworks:CreateLayer",
        "opsworks:DeleteLayer"
      ],
      "Resource": "arn:aws:opsworks:*:*:stack/2f18b4cb-4de5-4429-a149-ff7da9f0d8ee/"
    }
  ]
}
```

Pour plus d'informations, y compris des exemples de stratégie, consultez Gestion des autorisations de AWS OpsWorks Stacks en joignant une politique IAM.

> **Note**
>
> Une autre façon d'utiliser la politique IAM consiste à définir une condition qui limite l'accès à la pile aux employés possédant une adresse IP ou une plage d'adresses spécifiée. Par exemple, pour veiller à ce que les employés accèdent aux piles uniquement depuis l'intérieur de votre pare-feu d'entreprise, définissez une condition qui limite l'accès à votre plage d'adresses IP d'entreprise. Pour plus d'informations, consultez Conditions.

Bonnes pratiques : Gestion et déploiement des applications et des livres de recettes

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks déploie des applications et des livres de recettes sur chaque nouvelle instance à partir d'un référentiel distant. Au cours de la durée de vie d'une instance, vous devez souvent mettre à jour les applications ou les livres de recettes sur les instances en ligne de la pile pour ajouter des fonctions, corriger les bogues, etc. Il existe une grande variété de moyens pour gérer les applications et les livres de recettes d'une pile, mais l'approche que vous utilisez doit répondre aux exigences générales suivantes :

- Toutes les instances de pile en production doivent avoir le même code d'application et de livre de recettes personnalisé, avec des exceptions limitées telles que les tests A/B.
- Le déploiement d'une mise à jour ne doit pas interrompre le fonctionnement du site, même en cas de problème.

Cette section décrit les pratiques recommandées concernant la gestion et le déploiement d'applications et de livres de recettes personnalisés.

Rubriques

- Maintien de la cohérence
- Déploiement du code sur les instances en ligne

Maintien de la cohérence

En général, vous devez conserver un contrôle strict sur le code de l'application ou du livre de recettes qui s'exécute sur votre pile en production. En général, toutes les instances doivent exécuter la version actuellement approuvée du code. Les exceptions se produisent lors de la mise à jour de vos

applications ou livres de recettes, comme décrit plus tard, et lors de certains cas particuliers, comme l'exécution des tests A/B.

Le code de l'application et du livre de recettes est déployé depuis un référentiel source spécifié vers les instances de votre pile de deux façons :

- Lorsque vous démarrez une instance, AWS OpsWorks Stacks déploie automatiquement le code actuel de l'application et du livre de recettes sur l'instance.
- Pour les instances en ligne, vous devez déployer manuellement le code actuel de l'application ou du livre de recettes en exécutant une commande Deploy (pour les applications) ou une commande Update Custom Cookbooks (pour les livres de recettes).

Comme il existe deux mécanismes de déploiement, il est essentiel que vous gériez votre code source soigneusement afin d'éviter qu'un autre code ne s'exécute par inadvertance sur d'autres instances. Par exemple, si vous déployez des applications ou des livres de recettes à partir d'une branche principale de Git, AWS OpsWorks Stacks déploie le contenu de cette branche à ce moment-là. Si vous mettez à jour le code dans la branche principale, puis démarrez une nouvelle instance, cette instance possède une version plus récente du code que les instances plus anciennes. La version la plus récente peut même ne pas être approuvée pour la production.

Recommandation : Amazon S3 Archives

Pour garantir que toutes vos instances disposent de la version de code approuvée, nous vous recommandons de déployer vos applications et livres de recettes à partir d'une archive Amazon Simple Storage Service (Amazon S3). Cela garantit que le code est un artefact statique (un fichier .zip ou un autre fichier d'archive) qui doit être explicitement mis à jour. En outre, Amazon S3 est extrêmement fiable, de sorte que vous ne pourrez que rarement, voire jamais, accéder à l'archive. Pour garantir davantage de cohérence, versionnez explicitement chaque fichier d'archive en utilisant une convention de dénomination ou en utilisant le versionnement d'Amazon S3, qui fournit une piste d'audit et permet de revenir facilement à une version antérieure.

Par exemple, vous pouvez créer un pipeline de déploiement à l'aide d'un outil tel que Jenkins. Une fois que le code que vous souhaitez déployer a été validé et testé, créez un fichier d'archive et chargez-le sur Amazon S3. Tous les déploiements d'application ou mises à jour de livre de recettes installent le code dans ce fichier d'archives et chaque instance possède le même code.

Recommandation : référentiels Git ou Subversion

Si vous préférez utiliser un référentiel Git ou Subversion, ne le déployez pas à partir de la branche principale. Au lieu de cela, balisez la version approuvée et spécifiez cette version comme la source de l'application ou du livre de recettes.

Déploiement du code sur les instances en ligne

AWS OpsWorks Stacks ne déploie pas automatiquement le code mis à jour sur les instances en ligne. Vous devez effectuer cette opération manuellement, ce qui nécessite de relever les défis suivants :

- Déployer la mise à jour efficacement sans compromettre la capacité du site à gérer les demandes des clients pendant le processus de déploiement.
- Gérer un déploiement ayant échoué, en raison de problèmes liés à l'application ou aux livres de recettes déployés, ou de problèmes liés au processus de déploiement lui-même.

L'approche la plus simple consiste à exécuter une commande Deploy par défaut (pour les applications) ou une commande Update Custom Cookbooks (pour les livres de recettes), qui déploie la mise à jour sur chaque instance simultanément. Cette approche est simple et rapide, mais il n'y a aucune marge d'erreur. Si le déploiement échoue ou que le code mis à jour rencontre des problèmes, chaque instance de votre pile en production peut être affectée, ce qui perturbe ou désactive éventuellement votre site jusqu'à ce que vous puissiez résoudre le problème ou revenir à la version précédente.

Recommandation : utilisez une stratégie de déploiement robuste, qui permet aux instances exécutant l'ancienne version du code de poursuivre le traitement des demandes jusqu'à ce que vous ayez vérifié que le déploiement ait réussi et que vous puissiez transférer en toute confiance l'ensemble du trafic entrant vers la nouvelle version.

Les sections suivantes fournissent deux exemples de stratégies de déploiement robustes, suivis d'une discussion sur la manière de gérer une base de données principale pendant le déploiement. Pour être bref, elles décrivent les mises à jour des applications, mais vous pouvez utiliser des stratégies similaires pour les livres de recettes.

Rubriques

- Utilisation d'un déploiement roulant
- Utilisation de piles distinctes

- Gestion d'une base de données principale

Utilisation d'un déploiement roulant

Un déploiement roulant met à jour une application sur les instances serveur d'applications en ligne d'une pile en plusieurs phases. Avec chaque phase, vous mettez à jour un sous-ensemble des instances en ligne et vérifiez que la mise à jour a réussi avant de commencer la phase suivante. Si vous rencontrez des problèmes, les instances qui continuent d'exécuter l'ancienne version de l'application peuvent continuer à gérer le trafic entrant jusqu'à ce que vous ayez résolu les problèmes.

L'exemple suivant suppose que vous utilisiez la pratique recommandée en matière de répartition des instances serveur d'applications de votre pile entre plusieurs zones de disponibilité.

Pour effectuer un déploiement roulant

1. Sur la page Deploy App (Déployer l'application), choisissez Advanced (Avancé), sélectionnez une seule instance serveur d'applications et déployez l'application sur cette instance.

   Si vous souhaitez être prudent, vous pouvez supprimer l'instance de l'équilibreur de charge avant de déployer l'application. Les utilisateurs sont ainsi assurés de ne pas rencontrer l'application mise à jour jusqu'à ce que vous ayez vérifié qu'elle fonctionne correctement. Si vous utilisez Elastic Load Balancing, supprimez l'instance de l'équilibreur de charge à l'aide de la console Elastic Load Balancing, de la CLI ou d'un SDK.
2. Vérifiez que l'application mise à jour fonctionne correctement et que l'instance présente des métriques de performance acceptables.

   Si vous avez supprimé l'instance d'un équilibreur de charge Elastic Load Balancing, utilisez la console Elastic Load Balancing, la CLI ou un SDK pour la restaurer. La version d'application mise à jour gère désormais les requêtes utilisateur.
3. Déployez la mise à jour sur le reste des instances de la zone de disponibilité et vérifiez que les instances fonctionnent correctement et ont des métriques acceptables.
4. Répétez l'étape 3 pour les autres zones de disponibilité de la pile, une zone à la fois. Si vous voulez être particulièrement prudent, répétez les étapes 1 à 3.

> **Note**
>
> Si vous utilisez un équilibreur de charge Elastic Load Balancing, vous pouvez utiliser son bilan de santé pour vérifier que le déploiement a réussi. Cependant, définissez le

chemin d'accès ping vers une application qui contrôle les dépendances et vérifie que tout fonctionne correctement, pas vers un fichier statique qui confirme simplement que le serveur d'applications est en cours d'exécution.

Utilisation de piles distinctes

Une autre approche pour gérer les applications consiste à utiliser une pile distincte pour chaque phase du cycle de vie de l'application. Les différentes piles sont parfois appelées environnements. Cette organisation vous permet d'effectuer le développement et les tests sur des piles qui ne sont pas accessibles publiquement. Lorsque vous êtes prêt à déployer une mise à jour, basculez le trafic utilisateur à partir de la pile qui héberge la version actuelle de l'application vers la pile qui héberge la version mise à jour.

Rubriques

- Utilisation des piles de développement, des piles intermédiaires et des piles de production
- Utilisation d'une stratégie de déploiement blue-green

Utilisation des piles de développement, des piles intermédiaires et des piles de production

L'approche la plus courante utilise les piles suivantes.

Pile de développement

Utilisez une pile de développement pour des tâches telles que la mise en œuvre de nouvelles fonctionnalités ou la correction de bogues. Une pile de développement est essentiellement un prototype de pile de production, avec les mêmes couches, applications, ressources, etc., qui sont inclus dans votre pile de production. Comme, généralement, la pile de développement n'a pas à gérer la même charge que la pile de production, vous utilisez habituellement moins d'instances ou des instances plus petites.

Les piles de développement ne sont pas exposées publiquement ; vous contrôlez l'accès comme suit :

- Limitez l'accès réseau en configurant les règles de trafic entrant des groupes de sécurité du serveur d'applications ou de l'équilibreur de charge de façon à n'accepter que les demandes entrantes émanant d'adresses IP ou de plages d'adresses IP spécifiées.

Par exemple, limitez l'accès HTTP, HTTPS et SSH aux adresses de votre plage d'adresses d'entreprise.

- Contrôlez l'accès à la fonctionnalité de gestion des AWS OpsWorks piles Stacks en utilisant la page Permissions de la pile.

  Par exemple, accordez le niveau d'autorisations Manage à l'équipe de développement et le niveau Show à tous les autres employés.

Pile intermédiaire

Utilisez une pile intermédiaire pour tester et finaliser les candidats d'une pile en production mise à jour. Lorsque vous avez terminé le développement, créez une pile intermédiaire en clonant la pile de développement. Puis, exécutez votre suite de tests sur la pile intermédiaire et déployez les mises à jour sur la pile pour résoudre les problèmes qui surviennent.

Les piles de transit ne sont pas exposées au public ; vous contrôlez la pile et l'accès réseau de la même façon que vous le faites pour la pile de développement. Notez que lorsque vous clonez une pile de développement pour créer une pile intermédiaire, vous pouvez cloner les autorisations accordées par la gestion des autorisations de AWS OpsWorks Stacks. Cependant, le clonage n'affecte pas les autorisations accordées par les stratégies IAM des utilisateurs. Vous devez utiliser la console IAM, l'interface de ligne de commande ou un kit SDK pour modifier ces autorisations. Pour de plus amples informations, veuillez consulter Gestion des autorisations utilisateur.

Pile de production

La pile de production est la pile exposée au public et qui prend en charge votre application en cours. Lorsque la pile intermédiaire a passé les tests, vous la promouvez en pile de production et retirez l'ancienne pile de production. Pour obtenir un exemple montrant la façon de procéder, consultez Utilisation d'une stratégie de déploiement blue-green.

> **Note**
>
> Au lieu d'utiliser la console AWS OpsWorks Stacks pour créer des piles manuellement, créez un AWS CloudFormation modèle pour chaque pile. Cette méthode offre les avantages suivants :
>
> - Rapidité et commodité : lorsque vous lancez le modèle, il crée AWS CloudFormation automatiquement la pile, y compris toutes les instances requises.

- Cohérence : stockez le modèle pour chaque pile dans votre référentiel source afin de vous assurer que les développeurs utilisent la même pile dans le même but.

Utilisation d'une stratégie de déploiement blue-green

Une stratégie de déploiement bleu-vert est une solution courante pour utiliser efficacement des piles distinctes et déployer une mise à jour de l'application en production.

- L'environnement bleu est la pile de production, qui héberge l'application en cours.
- L'environnement vert est la pile intermédiaire, qui héberge l'application mise à jour.

Lorsque vous êtes prêt à déployer l'application mise à jour en production, vous basculez le trafic des utilisateurs de la pile bleue vers la pile verte, qui devient la nouvelle pile de production. Vous retirez alors l'ancienne pile bleue.

L'exemple suivant décrit comment effectuer un déploiement bleu vert avec des AWS OpsWorks stacks, en conjonction avec Route 53 et un pool d'équilibreurs de charge Elastic Load Balancing. Avant de procéder au basculement, vous devez vous assurer des points suivants :

- La mise à jour de l'application sur la pile verte a passé les tests avec succès et est prête pour la production.
- La pile verte est identique à la pile bleue, sauf qu'elle inclut l'application mise à jour et qu'elle n'est pas exposée au public.

  Les deux piles ont les mêmes autorisations, les mêmes nombre et type d'instance dans chaque couche, les mêmes configurations à date définie et à charge définie, etc.
- Toutes les instances 24/7 et les instances planifiées basés à date définie de la pile verte sont en ligne.
- Vous disposez d'un pool d'équilibreurs de charge Elastic Load Balancing qui peuvent être attachés dynamiquement à une couche dans l'une ou l'autre des piles et qui peuvent être préchauffés pour gérer le volume de trafic attendu.
- Vous avez utilisé la fonctionnalité de routage pondéré Route 53 pour créer un ensemble d'enregistrements dans une zone hébergée qui inclut vos équilibreurs de charge groupés.
- Vous avez attribué un poids différent de zéro à l'équilibreur de charge attaché à la couche serveur d'applications de votre pile bleue et un poids zéro aux équilibreurs de charge inutilisés. Cela garantit que l'équilibreur de charge de la pile bleue gère tout le trafic entrant.

Pour basculer les utilisateurs vers la pile verte

1. Attachez un des équilibreurs de charge inutilisés du pool à la couche du serveur d'application de la pile verte. Dans certains scénarios, comme lorsque vous prévoyez un trafic flash ou que vous ne pouvez pas configurer un test de charge de manière à augmenter progressivement le trafic, préchauffez l'équilibreur de charge afin de gérer le trafic attendu.
2. Une fois que toutes les instances de la pile verte ont passé avec succès le test de santé d'Elastic Load Balancing, modifiez les pondérations du jeu d'enregistrements Route 53 afin que l'équilibreur de charge de la pile verte ait une pondération différente de zéro et que le poids de l'équilibreur de charge de la pile bleue soit réduit en conséquence. Nous vous recommandons de commencer par ce que la pile verte gère un petit pourcentage de requêtes, peut-être 5 %, la pile bleue gérant le reste. Vous avez maintenant deux piles de production, avec la pile verte gérant certaines demandes entrantes et la pile bleue gérant le reste.
3. Surveillez les métriques de performance de la pile verte. Si elles sont acceptables, augmentez le poids de la pile verte afin qu'elle gère peut-être 10 % du trafic entrant.
4. Répétez l'étape 3 jusqu'à ce que la pile verte gère environ la moitié du trafic entrant. Tous les problèmes ayant dû être identifiés à ce stade, si la pile verte s'exécute correctement, vous pouvez terminer le processus en réduisant à zéro le poids de la pile bleue. La pile verte est désormais la nouvelle pile bleue et gère tout le trafic entrant.
5. Détachez l'équilibreur de charge de la couche serveur d'applications de l'ancienne pile bleue et retournez-le au pool.
6. Même si l'ancienne pile bleue ne gère plus les requêtes utilisateur, nous vous recommandons de la conserver quelque temps en cas de problèmes avec la nouvelle pile bleue. Dans ce cas, vous pouvez annuler la mise à jour en rétablissant la procédure et en redirigeant le trafic entrant vers l'ancienne pile bleue. Lorsque vous êtes sûr que la nouvelle pile bleue fonctionne correctement, arrêtez l'ancienne pile bleue.

Gestion d'une base de données principale

Si votre application dépend d'une base de données principale, vous devrez passer de l'ancienne application à la nouvelle. AWS OpsWorks Stacks prend en charge les options de base de données suivantes.

Couche Amazon RDS

Avec une couche Amazon Relational Database Service (Amazon RDS), vous créez l'instance de base de données RDS séparément, puis vous l'enregistrez dans votre stack. Vous pouvez

enregistrer une instance DB RDS avec une seule pile à la fois, mais vous pouvez basculer une instance DB RDS d'une pile vers une autre.

AWS OpsWorks Stacks installe un fichier contenant les données de connexion sur vos serveurs d'applications dans un format facilement utilisable par votre application. AWS OpsWorks Stacks ajoute également les informations de connexion à la base de données aux attributs de configuration et de déploiement de la pile, accessibles par des recettes. Vous pouvez également utiliser JSON pour fournir des données de connexion aux applications. Pour de plus amples informations, veuillez consulter Connexion à une base de données.

La mise à jour d'une application qui repose sur une base de données pose deux défis élémentaires :

- S'assurer que chaque transaction est correctement enregistrée pendant la transition tout en évitant également les conditions de concurrence entre les ancienne et nouvelle versions de l'application.
- Procéder à la transition d'une manière qui limite l'impact sur les performances de votre site, et réduit ou élimine les temps d'arrêt.

Lorsque vous utilisez les stratégies de déploiement décrites dans cette rubrique, vous ne pouvez pas simplement détacher la base de données de l'ancienne application et la rattacher à la nouvelle. Les deux versions de l'application s'exécutent en parallèle pendant la transition et doivent avoir accès aux mêmes données. Les lignes suivantes décrivent deux approches de la gestion de la transition, les deux présentant des avantages et posant des défis.

Approche 1 : Avoir deux applications qui se connectent à la même base de données

Avantages

- Il n'existe aucune interruption de service pendant la transition.

  Une application cesse progressivement d'accéder à la base de données, tandis que l'autre prend progressivement le contrôle.
- Vous n'avez pas à synchroniser les données entre les deux bases de données.

Défis

- Comme les deux applications accèdent à la même base de données, vous devez gérer l'accès pour éviter la perte ou corruption de données.
- Si vous avez besoin de migrer vers un nouveau schéma de base de données, l'ancienne version de l'application doit être en mesure d'utiliser le nouveau schéma.

Si vous utilisez des piles distinctes, cette approche convient probablement mieux à Amazon RDS, car l'instance n'est pas liée de façon permanente à une pile particulière et est accessible aux applications exécutées sur différentes piles. Cependant, comme vous ne pouvez pas enregistrer une instance de base de données RDS avec plus d'une pile à la fois, vous devez fournir les données de connexion aux deux applications, par exemple en utilisant JSON. Pour de plus amples informations, veuillez consulter Utilisation d'une recette personnalisée.

Si vous utilisez une mise à niveau progressive, l'ancienne et la nouvelle version de l'application sont hébergées sur la même pile. Vous pouvez donc utiliser une couche Amazon RDS ou MySQL.

Approche 2 : Offrir à chaque version de l'application sa propre base de données

Avantages

- Chaque version ayant sa propre base de données, les schémas n'ont pas à être compatibles.

Défis

- Synchroniser les données entre les deux bases de données pendant la transition sans perte ou corruption des données.
- S'assurer que votre procédure de synchronisation n'entraîne pas de temps d'arrêt significatifs ou ne dégrade pas considérablement les performances du site.

Si vous utilisez des piles distinctes, chacune d'elles a sa propre base de données. Si vous utilisez un déploiement roulant, vous pouvez attacher deux bases de données à la pile, une pour chaque application. Si l'ancienne application et l'application mise à jour n'ont pas de schémas de base de données compatibles, cette approche est préférable.

Recommandation : En général, nous recommandons d'utiliser une couche Amazon RDS comme base de données principale d'une application, car elle est plus flexible et peut être utilisée pour n'importe quel scénario de transition. Pour plus d'informations sur la gestion des transitions, consultez le guide de l'utilisateur Amazon RDS.

Empaquetage local des dépendances des livres de recettes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez utiliser Berkshelf pour empaqueter les dépendances de vos livres de recettes localement, télécharger le package sur Amazon S3 et modifier votre pile pour utiliser le package sur Amazon S3 comme source de livre de recettes. Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Les instructions pas à pas suivantes expliquent comment préempaqueter vos livres de recettes et leurs dépendances dans un fichier .zip, puis utiliser le fichier .zip en tant que source de livre de recettes pour les instances Linux de Stacks. AWS OpsWorks La première procédure explique comment empaqueter un livre de recettes. La deuxième procédure décrit comment empaqueter plusieurs livres de recettes.

Avant de commencer, installez le Kit de développement Chef (aussi appelé Chef DK), qui constitue un assortiment d'outils créés par la communauté Chef. Vous en aurez besoin pour utiliser l'outil de ligne de commande `chef`.

Création de packages de dépendances localement dans Chef 12

Dans Chef 12 Linux, Berkshelf n'est plus installé par défaut sur les instances de la pile. Nous vous recommandons d'installer et d'utiliser Berkshelf sur un ordinateur de développement local pour créer le package de vos dépendances de livres de recettes localement. Téléchargez votre package, avec les dépendances incluses, sur Amazon S3. Enfin, modifiez votre pile Chef 12 Linux pour utiliser le package chargé comme source de livre de recettes. Tenez compte des différences suivantes lorsque vous créez un package de livres de recettes dans Chef 12.

1. Sur l'ordinateur local, créez un livre de recettes en exécutant l'outil de ligne de commande `chef`.

   ```
   chef generate cookbook "server-app"
   ```

   Cette commande crée un livre de recettes, un fichier Berksfile, un fichier `metadata.rb` et un répertoire de recettes, puis les place dans un dossier ayant le même nom que le livre de recettes. L'exemple suivant montre la structure des éléments créés.

   ```
   server-app <-- the cookbook you've just created
   ```

```
    ### Berksfile
    ### metadata.rb
    ### recipes
```

2. Dans un éditeur de texte, modifiez le fichier Berksfile afin qu'il pointe vers des livres de recettes dont le livre de recettes `server-app` dépendra. Dans notre exemple, nous voulons que `server-app` dépende du livre de recettes `java` du supermarché Chef. Nous indiquons la version 1.50.0 ou une version mineure plus récente, mais vous pouvez indiquer n'importe quelle version publiée entre guillemets simples. Enregistrez les modifications, puis fermez le fichier.

```
source 'https://supermarket.chef.io'
cookbook 'java', '~> 1.50.0'
```

3. Modifiez le fichier `metadata.rb` pour ajouter la dépendance. Enregistrez les modifications, puis fermez le fichier.

```
depends 'java' , '~> 1.50.0'
```

4. Passez au répertoire de livres de recettes `server-app` que Chef a créé automatiquement, puis exécutez la commande `package` pour créer un fichier `tar` du livre de recettes. Si vous créez un package pour plusieurs livres de recettes, vous devez exécuter cette commande dans le répertoire racine dans lequel tous les livres de recettes sont stockés. Pour créer un package pour un seul livre de recettes, exécutez cette commande au niveau du répertoire du livre de recettes. Dans cet exemple, nous exécutons cette commande dans le répertoire `server-app`.

```
berks package cookbooks.tar.gz
```

   La sortie se présente comme suit : Le fichier `tar.gz` est créé dans votre répertoire local.

```
Cookbook(s) packaged to /Users/username/tmp/berks/cookbooks.tar.gz
```

5. Dans le AWS CLI, téléchargez le package que vous venez de créer sur Amazon S3. Notez la nouvelle URL du package du livre de recettes après sont chargement dans S3. Vous aurez besoin de cette URL pour les paramètres de votre pile.

```
aws s3 cp cookbooks.tar.gz s3://bucket-name/
```

   La sortie se présente comme suit :

```
upload: ./cookbooks.tar.gz to s3://bucket-name/cookbooks.tar.gz
```

6. Dans AWS OpsWorks Stacks, modifiez votre pile pour utiliser le package que vous avez téléchargé comme source du livre de recettes.

   a. Définissez le paramètre Use custom Chef cookbooks (Utiliser les livres de recettes Chef personnalisés) sur Yes (Oui).

   b. Définissez Repository type (Type de référentiel) sur S3 Archive (Archive S3).

   c. Dans le champ Repository URL (URL du référentiel), collez l'URL du package du livre de recettes que vous avez chargé à l'étape 5.

   Enregistrez les modifications apportées à la pile.

Empaquetage local des dépendances pour un seul livre de recettes

1. Sur l'ordinateur local, créez un livre de recettes à l'aide de l'outil de ligne de commande chef :

   ```
   chef generate cookbook "server-app"
   ```

   Cette commande crée un livre de recettes et un fichier Berksfile, puis les place dans un dossier ayant le même nom que le livre de recettes.

2. Changez pour le répertoire de livres de recettes que Chef a créé automatiquement, puis empaquetez l'ensemble en exécutant la commande suivante :

   ```
   berks package cookbooks.tar.gz
   ```

   Le résultat se présente comme suit :

   ```
   Cookbook(s) packaged to /Users/username/tmp/berks/cookbooks.tar.gz
   ```

3. Dans le AWS CLI, téléchargez le package que vous venez de créer sur Amazon S3 :

   ```
   aws s3 cp cookbooks.tar.gz s3://bucket-name/
   ```

   Le résultat se présente comme suit :

```
upload: ./cookbooks.tar.gz to s3://bucket-name/cookbooks.tar.gz
```

4. Dans AWS OpsWorks Stacks, modifiez votre pile pour utiliser le package que vous avez téléchargé comme source du livre de recettes.

Empaquetage local des dépendances pour plusieurs livres de recettes

Cet exemple crée deux livres et empaquette les dépendances.

1. Sur l'ordinateur local, exécutez les commandes `chef` suivantes pour générer deux livres de recettes :

   ```
   chef generate cookbook "server-app"
   chef generate cookbook "server-utils"
   ```

   Dans cet exemple, comme le livre de recettes server-app effectue les configurations Java, nous avons besoin d'ajouter une dépendance sur Java.

2. Modifiez `server-app/metadata.rb` pour ajouter une dépendance sur le livre de recettes Java de la communauté :

   ```
   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. Dites à Berkshelf ce qui doit être empaqueté en modifiant comme suit le fichier Berksfile dans le répertoire racine des livres de recettes :

   ```
   source "https://supermarket.chef.io"
   cookbook "server-app", path: "./server-app"
   cookbook "server-utils", path: "./server-utils"
   ```

   La structure de votre fichier se présente comme suit :

   ```
   ..
   ```

```
    ### Berksfile
    ### server-app
    ### server-utils
```

4. Enfin, créez un package zip, chargez-le sur Amazon S3 et modifiez votre pile AWS OpsWorks Stacks pour utiliser la nouvelle source du livre de recettes. Pour ce faire, suivez les étapes 2 à 4 de Empaquetage local des dépendances pour un seul livre de recettes.

Ressources supplémentaires

Pour plus d'informations sur la création de packages de dépendances de livres de recettes, consultez les rubriques suivantes.

- Comment empaqueter les dépendances des livres de recettes localement avec Berkshelf sur le blog AWS DevOps
- Linux Chef 12 avec Berkshelf sur les forums AWS OpsWorks
- Berkshelf dans Chef 12 sur les forums AWS OpsWorks
- Installation de livres de recettes personnalisés dans le présent guide
- Référentiels de livres de recettes dans le présent guide

Piles

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

La pile est l'entité AWS OpsWorks Stacks de niveau supérieur. Elle représente un ensemble d'instances que vous voulez gérer de manière collective, généralement parce qu'elles ont un objectif commun tel que le traitement des applications PHP. En plus de servir un conteneur, une pile gère les tâches qui s'appliquent au groupe d'instances dans son ensemble, par exemple la gestion des applications et des livres de recettes.

Par exemple, une pile dont le but est de servir des applications web peut ressembler à ce qui suit :

- Un ensemble d'instances de serveur d'applications, chacune d'entre elles gérant une partie du trafic entrant.
- Une instance de l'équilibreur de charge, qui prend le trafic entrant et le répartit sur les serveurs d'applications.
- Une instance de base de données, qui joue le rôle de magasin de données principal pour les serveurs d'applications.

Une pratique courante consiste à avoir plusieurs piles qui représentent différents environnements. Un ensemble classique de piles se compose de :

- Une pile de développement utilisée par les développeurs pour ajouter des fonctionnalités, résoudre les bogues et effectuer d'autres tâches de développement et de maintenance.
- Une pile intermédiaire pour vérifier les mises à jour ou les correctifs avant de les exposer publiquement.
- Une pile de production, c'est-à-dire la version destinée au public qui gère les requêtes entrantes des utilisateurs.

Cette section décrit les bases de l'utilisation des piles.

Rubriques

- Migrer des piles d'Amazon EC2 Classic vers un VPC
- Créer une pile
- Running a Stack in a VPC
- Mise à jour d'une pile
- Cloner une pile
- Commandes AWS OpsWorks Run Stacks Stack
- Utilisation du JSON personnalisé
- Suppression d'une pile

Migrer des piles d'Amazon EC2 Classic vers un VPC

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette rubrique décrit comment procéder pour procéder à la migration d'une AWS OpsWorks Stacks pile de la plateforme réseau Amazon EC2 Classic vers un réseau Amazon Virtual Private Cloud (Amazon VPC).

Si vous avez créé votre AWS compte avant le 4 décembre 2013, vous pouvez bénéficier de la prise en charge de EC2 -Classic dans certaines régions. AWS Certaines EC2 ressources et fonctions Amazon, comme la mise en réseau améliorée et des types d'instance plus récents, requièrent un cloud privé virtuel (VPC). Certaines ressources peuvent être partagées entre EC2 -Classic et un VPC, contrairement à d'autres ressources. Pour éviter toute interruption de service, nous vous recommandons de migrer vos AWS OpsWorks Stacks piles vers un VPC.

Rubriques

- Prérequis
- Migrer une AWS OpsWorks Stacks pile vers un VPC
- Consultez aussi

Prérequis

Avant de commencer, vous devez disposer d'un VPC répondant aux exigences de AWS OpsWorks Stacks configuration. Pour configurer des sous-réseaux privés dans votre VPC AWS OpsWorks Stacks pour, Running a Stack in a VPC consultez ce guide. Vous pouvez créer un VPC personnalisé à l'aide de la console de gestion Amazon VPC. Pour plus d'informations, consultez la section Configurations VPCs et sous-réseaux de l'assistant de console Amazon VPC dans le guide de l'utilisateur Amazon Virtual Private Cloud.

Pour poursuivre la migration, vous aurez besoin de l'ID du VPC et de l'ID de sous-réseau que vous souhaitez utiliser.

Migrer une AWS OpsWorks Stacks pile vers un VPC

Tout d'abord, clonez une pile EC2 -Classic existante à l'aide de la AWS OpsWorks Stacks console ou de l'API. Déplacez ensuite les ressources de la pile existante vers la nouvelle pile. Démarrez les nouvelles instances de la pile clonée et déployez les applications. Vérifiez que la nouvelle pile fonctionne. Enfin, supprimez les ressources EC2 -Classic de la pile EC2 -Classic, puis supprimez l'ancienne pile.

1. Clonez votre stack EC2 -Classic existant dans votre VPC. Le clonage de la pile copie les paramètres de la pile, les couches, les applications, les utilisateurs et les autorisations des utilisateurs vers la nouvelle pile. Pour plus d'informations sur le clonage d'une pile, veuillez consulter Cloner une pile dans ce guide.

   Vous pouvez également cloner une pile à l'aide de l' AWS OpsWorks Stacks API. Lorsque vous clonez une pile à l'aide du AWS CLI ou AWS SDKs, définissez la valeur du `VpcId` paramètre sur l'ID du VPC dans lequel vous l'avez créé. Prérequis Pour plus d'informations, consultez CloneStack dans la Référence d'API AWS OpsWorks Stacks .
2. Créez de nouvelles instances dans les couches de la pile clonée. Assurez-vous de spécifier l'ID du sous-réseau dans Prérequis lequel vous l'avez créé. Pour plus d'informations sur la création d'instances dans une pile, veuillez consulter Ajout d'une instance à une couche dans ce guide.
3. Migrez vos ressources classiques, telles que les groupes EC2 de sécurité, les équilibreurs de charge Elastic Load Balancing et les adresses IP Elastic vers votre VPC, puis associez-les à la pile clonée. Pour plus d'informations, consultez Migrer vos ressources vers un VPC dans le guide de EC2 l'utilisateur Amazon.
4. Enregistrez les volumes Amazon EBS et les instances Amazon RDS auprès de la pile clonée. Pour plus d'informations sur l'enregistrement de ressources auprès d'une pile, veuillez consulter Enregistrement des ressources auprès d'une pile dans ce guide.

   Les volumes Amazon EBS ne sont pas associés à un VPC, et vous pouvez les utiliser sur plusieurs instances à la EC2 fois dans les piles classiques et dans les piles d'un VPC. Vous pouvez enregistrer des instances Amazon RDS dans EC2 -Classic avec des piles EC2 -Classic et des piles dans un VPC.
5. Démarrez des instances dans la pile clonée, puis déplacez un petit pourcentage de vos charges de travail vers la pile clonée. Par exemple, déplacez un faible pourcentage du trafic vers les

équilibreurs de charge Elastic Load Balancing de la pile clonée. Si vous utilisez Amazon Route 53, consultez Routage du trafic vers un équilibreur de charge ELB (français non garanti) dans le Manuel du développeur Amazon Route 53 (français non garanti).

N'acheminez qu'un faible pourcentage du trafic jusqu'à ce que vous soyez certain que la nouvelle pile est fonctionnelle et prend en charge vos applications. Laissez la nouvelle pile fonctionner avec un faible pourcentage de trafic pendant une période d'essai, par exemple une semaine. Après avoir vérifié que la nouvelle pile fonctionne, acheminez le trafic restant vers la pile.

6. Une fois que vous êtes certain que la pile clonée fonctionne, déplacez le reste de votre trafic de production ou de vos charges de travail vers la pile clonée. Vous pouvez désormais arrêter les instances de la pile EC2 -Classic. Nous vous recommandons de garder l'ancienne pile disponible pendant plusieurs semaines, afin de pouvoir transférer les charges de travail vers l'ancienne pile si des problèmes surviennent avec la nouvelle pile dans les semaines suivant la migration.
7. Lorsque la nouvelle pile fonctionne depuis plusieurs semaines, supprimez les instances de la pile EC2 -Classic. Pour plus d'informations sur la suppression d'instances, veuillez consulter Supprimer des AWS OpsWorks instances de Stacks dans ce guide.

   > **Important**
   >
   > N'utilisez pas la EC2 console ou l'API Amazon pour arrêter ou supprimer AWS OpsWorks des instances.

8. Supprimez les applications de la pile EC2 -Classic. Pour plus d'informations sur la suppression d'applications, consultez Pour supprimer l'application de la pile.
9. Supprimez la pile EC2 -Classic. Pour plus d'informations sur la suppression d'une pile, veuillez consulter Suppression d'une pile dans ce guide.

Consultez aussi

- Migrer d' EC2-Classic vers un VPC
- Guide de débogage et dépannage
- Running a Stack in a VPC

Créer une pile

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Pour créer une nouvelle pile, dans le tableau de bord AWS OpsWorks Stacks, cliquez sur Ajouter une pile. Vous pouvez ensuite utiliser la page Ajouter une pile pour configurer la pile. Lorsque vous avez terminé, cliquez sur Ajouter une pile.

Rubriques

- Choisissez le type de pile à créer
- Options de base
- Options avancées

Choisissez le type de pile à créer

Avant de créer une pile, vous devez décider du type de pile que vous souhaitez créer. Consultez le tableau suivant pour obtenir de l'aide.

Si vous souhaitez créer...	Créez ce type de pile si vous voulez...	Pour découvrir comment, suivez les instructions ci-après :
Un exemple de pile	Découvrez les bases d'AWS OpsWorks avec une pile Chef 12 basée sur Linux et un exemple d'application Node.js.	Mise en route : exemple
Pile basée Chef 12 basée sur Linux	Créez une pile basée sur Linux qui utilise la dernière	Mise en route : Linux

Si vous souhaitez créer...	Créez ce type de pile si vous voulez...	Pour découvrir comment, suivez les instructions ci-après :
	version de Chef prise en charge par AWS. OpsWorks Sélectionnez cette option si vous êtes un utilisateur avancé de Chef et que vous souhaitez bénéficier de la grande sélection de livres de recettes de la communauté ou écrire vos propres livres de recettes personnalisés. Pour de plus amples informations, veuillez consulter Chef 12 Linux.	
Pile Chef 12.2 basée sur Windows	Créez une pile basée sur Windows.	Mise en route : Windows
Pile basée Chef 11.10 basée sur Linux	Créez cette pile si votre organisation nécessite l'utilisation de Chef 11.10 avec Linux pour la compatibilité descendante.	Mise en route des piles Linux Chef 11

Options de base

La page Ajouter une pile présente les options de base suivantes.

Nom de la pile

(Obligatoire) Nom utilisé pour identifier la pile dans la console AWS OpsWorks Stacks. Le nom ne doit pas nécessairement être unique. AWS OpsWorks Stacks génère également un identifiant de pile, qui est un GUID identifiant de manière unique la pile. Par exemple, avec les commandes de l'interface de ligne de commande AWS, telles que update-stack, vous utilisez l'ID de la pile pour identifier la pile particulière. Une fois que vous avez créé une pile, vous pouvez trouver son ID en

choisissant Stack dans le volet de navigation, puis en choisissant Stack Setting. L'identifiant est étiqueté OpsWorks ID.

Région

(Obligatoire) La région AWS où les instances seront lancées.

VPC

(Facultatif) L'ID du VPC dans lequel la pile doit être lancée. Toutes les instances seront lancées dans ce VPC et vous ne pourrez pas modifier l'ID ultérieurement.

- Si votre compte prend en charge la EC2 version classique, vous pouvez spécifier Aucun VPC (valeur par défaut) si vous ne souhaitez pas utiliser de VPC.

  Pour plus d'informations sur EC2 Classic, consultez la section Plateformes prises en charge.
- Si votre compte ne prend pas en charge EC2 Classic, vous devez spécifier un VPC.

  Le paramètre par défaut est VPC par défaut, qui combine la facilité d'utilisation de EC2 Classic avec les avantages des fonctionnalités réseau VPC. Si vous souhaitez exécuter votre pile dans un VPC standard, vous devez la créer à l'aide de la console, de l'API ou de l'interface de ligne de commande du VPC. Pour plus d'informations sur la création d'un VPC pour une pile AWS OpsWorks Stacks, consultez Running a Stack in a VPC. Pour plus d'informations générales, consultez Amazon Virtual Private Cloud.

Zone de disponibilité par défaut/sous-réseau par défaut

(Facultatif) Ce paramètre varie selon votre choix de créer une pile dans un VPC :

- Si votre compte prend en charge la EC2 version classique et que vous définissez VPC sur Aucun VPC, ce paramètre est intitulé Zone de disponibilité par défaut, qui indique la zone de disponibilité AWS par défaut dans laquelle les instances seront lancées.
- Si votre compte n'est pas compatible avec EC2 Classic ou si vous choisissez de spécifier un VPC, ce champ est intitulé Sous-réseau par défaut, qui indique le sous-réseau par défaut dans lequel les instances seront lancées. Vous pouvez lancer une instance dans d'autres sous-réseaux en remplaçant cette valeur lorsque vous créez l'instance. Chaque sous-réseau est associé à une zone de disponibilité.

Vous pouvez demander à AWS OpsWorks Stacks de lancer une instance dans une autre zone de disponibilité ou un autre sous-réseau en remplaçant ce paramètre lors de la création de l'instance.

Pour plus d'informations sur l'exécution d'une pile dans un VPC;, consultez Running a Stack in a VPC.

Système d'exploitation par défaut

(Facultatif) Le système d'exploitation qui est installé par défaut sur chaque instance. Vous avez les options suivantes :

- L'un des systèmes d'exploitation Linux intégrés.
- Microsoft Windows Server 2012 R2.
- Une AMI personnalisée basée sur l'un des systèmes d'exploitation pris en charge.

  Si vous sélectionnez Use custom AMI (Utiliser une AMI personnalisée), le système d'exploitation est déterminé par une AMI personnalisée que vous spécifiez lorsque vous créez des instances. Pour de plus amples informations, veuillez consulter Utilisation de Custom AMIs.

Pour plus d'informations sur les systèmes d'exploitation disponibles, consultez AWS OpsWorks Systèmes d'exploitation Stacks.

> **Note**
>
> Vous pouvez remplacer le système d'exploitation par défaut lorsque vous créez une instance. Toutefois, vous ne pouvez pas remplacer un système d'exploitation Linux pour spécifier Windows ou un système d'exploitation Windows pour spécifier Linux.

Clé SSH par défaut

(Facultatif) Une paire de EC2 clés Amazon provenant de la région de la pile. La valeur par défaut est none. Si vous spécifiez une paire de clés, AWS OpsWorks Stacks installe la clé publique sur l'instance.

- Avec les instances Linux, vous pouvez utiliser la clé privée avec un client SSH pour vous connecter aux instances de la pile.

  Pour de plus amples informations, veuillez consulter Connexion avec SSH.
- Avec les instances Windows, vous pouvez utiliser la clé privée avec la EC2 console Amazon ou la CLI pour récupérer le mot de passe administrateur d'une instance.

  Vous pouvez ensuite utiliser ce mot de passe avec un client RDP pour vous connecter à l'instance en tant qu'Administrateur. Pour de plus amples informations, veuillez consulter Connexion avec RDP.

Pour plus d'informations sur la façon de gérer les clés SSH, consultez Gestion de l'accès SSH.

> **Note**
>
> Vous pouvez remplacer ce paramètre en spécifiant une paire de clés différente, ou aucune paire de clés, lorsque vous créez une instance.

Version de Chef

Montre la version de Chef que vous avez choisie.

Pour plus d'informations sur les versions de Chef, consultez Versions Chef.

Utiliser les livres de recettes Chef personnalisés

S'il convient d'installer vos livres de recettes Chef personnalisés sur les instances de la pile.

Pour Chef 12, le paramètre par défaut est Yes. Pour Chef 11, le paramètre par défaut est Non. L'option Oui affiche plusieurs paramètres supplémentaires qui fournissent à AWS OpsWorks Stacks les informations dont il a besoin pour déployer les livres de recettes personnalisés de son référentiel vers les instances de la pile, telles que l'URL du référentiel. Les détails dépendent du référentiel utilisé pour vos livres de recettes. Pour de plus amples informations, veuillez consulter Installation de livres de recettes personnalisés.

Couleur de la pile

(Facultatif) La teinte utilisée pour représenter la pile sur la console AWS OpsWorks Stacks. Vous pouvez utiliser différentes couleurs pour différentes piles afin de faciliter la distinction, par exemple entre les piles de développement, intermédiaires et de production.

Balises de pile

Vous pouvez appliquer des balises au niveau de la pile et de la couche. Lorsque vous créez une balise, vous appliquez la balise à chaque ressource au sein de la structure balisée. Par exemple, si vous appliquez une balise à une pile, vous l'appliquez à chaque couche, et au sein de chaque couche, à chaque instance, volume Amazon EBS ou équilibreur de charge Elastic Load Balancing de la couche. Pour plus d'informations sur la façon d'activer vos tags et de les utiliser pour suivre et gérer les coûts de vos ressources AWS OpsWorks Stacks, consultez les sections Utilisation des balises de répartition des coûts et activation des balises de répartition des coûts définies par l'utilisateur dans le guide de l'utilisateur de Billing and Cost Management. Pour plus d'informations sur le balisage dans AWS OpsWorks Stacks, consultez. Balises

Options avancées

Pour les paramètres avancés, cliquez sur Advanced **>>** pour afficher les sections Advanced options et Security.

La section Advanced options a les options suivantes :

Taille du périphérique racine par défaut

Détermine le type de stockage à utiliser pour le volume racine de l'instance. Pour plus d'informations, consultez Stockage.

- Les pile Linux utilisent un volume racine basé sur des volumes Amazon EBS par défaut, mais vous pouvez également spécifier un volume racine basé sur un stockage d'instance.
- Les piles Windows Stacks doivent utiliser un volume racine d'Amazon EBS.

Rôle IAM

(Facultatif) Rôle AWS Identity and Access Management (IAM) de la pile, que AWS OpsWorks Stacks utilise pour interagir avec AWS en votre nom.

Profil d'instance IAM par défaut

(Facultatif) Rôle IAM par défaut à associer aux EC2 instances Amazon de la pile. Ce rôle donne des autorisations aux applications exécutées sur les instances de la pile pour accéder aux ressources AWS telles que les compartiments S3.

- Pour accorder des autorisations spécifiques aux applications, choisissez un profil d'instance existant (rôle) ayant les stratégies appropriées.
- Au départ, le rôle du profil n'accorde aucune autorisation, mais vous pouvez utiliser la console IAM, l'API ou la CLI pour associer les politiques appropriées. Pour de plus amples informations, veuillez consulter Spécification des autorisations pour les applications exécutées sur EC2 des instances.

Région du point de terminaison de l'API

La valeur de ce paramètre vient de la région choisie dans les paramètres de base de la pile. Vous pouvez choisir parmi les points de terminaison régionaux suivants.

- US East (N. Virginia) Region
- US East (Ohio) Region
- US West (Oregon) Region
- Région US West (N. California)

- Région du Canada (Centre) (API uniquement) ; non disponible pour les piles créées dans le AWS Management Console
- Région Asie-Pacifique (Mumbai)
- Région Asia Pacific (Singapore)
- Région Asie-Pacifique (Sydney)
- Région Asia Pacific (Tokyo)
- Région Asia Pacific (Seoul)
- Région Europe (Frankfurt)
- Région Europe (Irlande)
- Région Europe (Londres)
- Région Europe (Paris)
- Région Amérique du Sud (São Paulo)

Les piles qui sont créées dans un point de terminaison de l'API ne sont pas disponibles dans un autre point de terminaison de l'API. Les utilisateurs de AWS OpsWorks Stacks étant également spécifiques à une région, si vous souhaitez que les utilisateurs de AWS OpsWorks Stacks de l'une de ces régions de point de terminaison gèrent les piles d'une autre région de point de terminaison, vous devez importer les utilisateurs vers le point de terminaison auquel les piles sont associées. Pour plus d'informations sur l'importation d'utilisateurs, voir Importation d'utilisateurs dans AWS OpsWorks Stacks.

Thème du nom d'hôte

(Facultatif) Une chaîne qui est utilisée pour générer un nom d'hôte par défaut pour chaque instance. La valeur par défaut est Dépendant de la couche, qui utilise le nom court de la couche d'instance et ajoute un numéro unique à chaque instance. Par exemple, la racine du thème de l'Équilibreur de charge dépendant du rôle est « lb ». La première instance que vous ajoutez à la couche est nommée « lb1 », la deuxième « lb2 » et ainsi de suite.

OpsWorks Version de l'agent

(Facultatif) S'il faut mettre à jour automatiquement l'agent AWS OpsWorks Stacks lorsqu'une nouvelle version est disponible, ou s'il faut utiliser une version d'agent spécifiée et la mettre à jour manuellement. Cette fonctionnalité est disponible sur les piles Chef 11.10 et Chef 12. Le paramètre par défaut est Mise à jour manuelle, défini sur la dernière version de l'agent.

AWS OpsWorks Stacks installe un agent sur chaque instance qui communique avec le service et gère des tâches telles que le lancement d'exécutions de Chef en réponse à des événements

du cycle de vie. Cet agent est régulièrement mis à jour. Vous avez deux options pour spécifier la version de l'agent de votre pile.

- Mise à jour automatique : AWS OpsWorks Stacks installe automatiquement chaque nouvelle version de l'agent sur les instances de la pile.
- Mise à jour manuelle : AWS OpsWorks Stacks installe la version de l'agent spécifiée sur les instances de la pile.

  AWS OpsWorks Stacks publie un message sur la page de la pile lorsqu'une nouvelle version de l'agent est disponible, mais ne met pas à jour les instances de la pile. Pour mettre à jour l'agent, vous devez mettre à jour manuellement les paramètres de la pile pour spécifier une nouvelle version de l'agent. AWS OpsWorks Stacks mettra alors à jour les instances de la pile.

Vous pouvez remplacer le paramètre de version de l'OpsWorks agent par défaut pour une instance donnée en mettant à jour sa configuration. Dans ce cas, le paramètre de l'instance a priorité. Par exemple, supposons que le paramètre par défaut soit Mise à jour automatique, mais que vous choisissiez de spécifier Mise à jour manuelle pour une instance spécifique. Lorsque AWS OpsWorks Stacks publie une nouvelle version d'agent, toutes les instances de la pile sont automatiquement mises à jour, à l'exception de celle qui est définie sur Mise à jour manuelle. Pour installer une nouvelle version de l'agent sur cette instance, vous devez manuellement mettre à jour sa configuration et spécifier une nouvelle version.

> **Note**
>
> La console affiche les numéros abrégés de version de l'agent. Pour voir les numéros de version complets, appelez la describe-agent-versionscommande AWS CLI ou les méthodes API ou SDK équivalentes. Elles retournent les numéros de version complets pour les versions disponibles de l'agent.

JSON personnalisé

(Facultatif) Un ou plusieurs attributs personnalisés, au format de structure JSON. Ces attributs sont fusionnées dans les attributs de configuration et de déploiement de la pile qui sont installé sur chaque instance et peuvent être utilisés par des recettes. Vous pouvez utiliser un JSON personnalisé, par exemple pour personnaliser les paramètres de configuration en remplaçant les attributs intégrés qui spécifient les paramètres par défaut. Pour de plus amples informations, veuillez consulter Utilisation du JSON personnalisé.

La sécurité comporte une option, Utiliser les groupes de OpsWorks sécurité, qui vous permet de spécifier s'il faut associer les groupes de sécurité intégrés de AWS OpsWorks Stacks aux couches de la pile.

AWS OpsWorks Stacks fournit un ensemble standard de groupes de sécurité intégrés (un par couche) associés aux couches. Utiliser des groupes OpsWorks de sécurité vous permet de fournir à la place vos propres groupes de sécurité personnalisés. Pour de plus amples informations, veuillez consulter Utilisation des groupes de sécurité.

Les paramètres d'utilisation des groupes de OpsWorks sécurité sont les suivants :

- Oui. AWS OpsWorks Stacks associe automatiquement le groupe de sécurité intégré approprié à chaque couche (paramètre par défaut).

  Vous pouvez associer des groupes de sécurité supplémentaires à une couche après l'avoir créée, mais vous ne pouvez pas supprimer le groupe de sécurité intégré.
- Non, AWS OpsWorks Stacks n'associe pas les groupes de sécurité intégrés aux couches.

  Vous devez créer les groupes EC2 de sécurité appropriés et associer un groupe de sécurité à chaque couche que vous créez. Cependant, il reste possible d'associer manuellement un groupe de sécurité préintégré à une couche lors de sa création ; les groupes de sécurité personnalisés ne sont nécessaires que pour les couches requérant des paramètres personnalisés.

Remarques :

- Si l'option Utiliser les groupes de OpsWorks sécurité est définie sur Oui, vous ne pouvez pas restreindre les paramètres d'accès aux ports d'un groupe de sécurité par défaut en ajoutant un groupe de sécurité plus restrictif à une couche. Avec plusieurs groupes de sécurité, Amazon EC2 utilise les paramètres les plus permissifs. En outre, vous ne pouvez pas créer de paramètres plus restrictifs en modifiant la configuration du groupe de sécurité intégré. Lorsque vous créez une pile, AWS OpsWorks Stacks remplace les configurations des groupes de sécurité intégrés par les paramètres standard, de sorte que toutes les modifications que vous apporterez seront perdues lors de la prochaine création d'une pile. Si une couche nécessite des paramètres de groupe de sécurité plus restrictifs que le groupe de sécurité intégré, définissez Utiliser les groupes de OpsWorks sécurité sur Non, créez des groupes de sécurité personnalisés avec vos paramètres préférés et attribuez-les aux couches lors de leur création.
- Si vous supprimez accidentellement un groupe de sécurité AWS OpsWorks Stacks et que vous souhaitez le recréer, il doit être exactement le double de l'original, y compris la majuscule du nom

du groupe. Au lieu de recréer le groupe manuellement, nous vous recommandons de faire en sorte qu' AWS OpsWorks Stacks effectue cette tâche pour vous. Créez simplement une nouvelle pile dans la même région AWS (et un VPC, le cas échéant AWS OpsWorks). Stacks recréera automatiquement tous les groupes de sécurité intégrés, y compris celui que vous avez supprimé. Vous pouvez ensuite supprimer la pile si vous n'en avez plus l'utilisation ; les groupes de sécurité demeureront.

Running a Stack in a VPC

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez contrôler l'accès utilisateur aux instances d'une pile en la créant dans un Virtual Private Cloud (VPC). Par exemple, il se peut que vous ne vouliez pas que les utilisateurs accèdent directement aux serveurs d'applications ou aux bases de données de votre pile, et, qu'à la place, tout le trafic public soit acheminé via un Elastic Load Balancer.

La procédure de base pour exécuter une pile dans un VPC est la suivante :

1. Créez un VPC correctement configuré à l'aide de la console ou de l'API Amazon VPC, ou d'un modèle. AWS CloudFormation
2. Spécifiez l'ID VPC lorsque vous créez la pile.
3. Lancez les instances de la pile dans le sous-réseau approprié.

Ce qui suit décrit brièvement le VPCs fonctionnement de AWS OpsWorks Stacks.

> **Important**
>
> Si vous utilisez la fonctionnalité de point de terminaison d'un VPC, sachez que chaque instance de la pile doit être en mesure d'effectuer les actions suivantes à partir d'Amazon Simple Storage Service (Amazon S3) :

- Installer l'agent de l'instance.
- Installez les ressources, tels que Ruby.
- Charger les journaux d'exécution Chef.
- Récupérer les commandes de pile.

Pour activer ces actions, vous devez vous assurer que les instances de la pile ont accès aux compartiments suivants qui correspondent à la région de la pile. Sinon, les actions précédentes échouent.
Pour Chef 12 Linux et Chef 12.2 Windows, les compartiments sont les suivants.

Compartiments d'agents	Compartiments de ressources	Compartiments de journaux	Compartiments DNA
• opsworks-instance-agent-sa-east-1 • opsworks-instance-agent-ap-south-1 • opsworks-instance-agent-ap-northeast-1 • opsworks-instance-agent-ap-northeast-2 • opsworks-instance-agent-ap-southeast-1 • opsworks-instance-agent-ap-southeast-2 • opsworks-instance-agent-ca-central-1 • opsworks-instance-agent-eu-central-1 • opsworks-instance-agent-eu-west-1 • opsworks-instance-agent-eu-ouest-2	• opsworks-instance-assets-us-est-2 • opsworks-instance-assets-us-east-1 • opsworks-instance-assets-ap-south-1 • opsworks-instance-assets-ap-northeast-1 • opsworks-instance-assets-ap-northeast-2 • opsworks-instance-assets-ap-southeast-1 • opsworks-instance-assets-ap-southeast-2 • opsworks-instance-assets-ca-central-1 • opsworks-instance-assets-eu-central-1 • opsworks-instance-assets-eu-west-1	• opsworks-us-east-2 journaux • opsworks-us-east-1 journal • opsworks-ap-south-1 journal • opsworks-ap-northeast-1 journal • opsworks-ap-northeast-2 journaux • opsworks-ap-southeast-1 journal • opsworks-ap-southeast-2 journaux • opsworks-ca-central-1 journal • opsworks-eu-central-1 journal • opsworks-eu-west-1 journal • opsworks-eu-west-2 journaux • opsworks-eu-west-3 journaux • opsworks-sa-east-1 journal	• opsworks-us-east-2-ADN • opsworks-us-east-1-ADN • opsworks-ap-south-1-ADN • opsworks-ap-northeast-1-ADN • opsworks-ap-northeast-2-ADN • opsworks-ap-southeast-1-ADN • opsworks-ap-southeast-2-ADN • opsworks-ca-central-1-ADN • opsworks-eu-central-1-ADN • opsworks-eu-west-1-ADN • opsworks-eu-west-2-ADN • opsworks-eu-west-3-ADN • opsworks-sa-east-1-ADN • opsworks-us-west-1-ADN • opsworks-us-west-2-ADN

Compartiments d'agents	Compartiments de ressources	Compartiments de journaux	Compartiments DNA
• opsworks-instance-agent-eu-west-3 • opsworks-instance-agent-us-east-1 • opsworks-instance-agent-us-est-2 • opsworks-instance-agent-us-west-1 • opsworks-instance-agent-us-ouest-2	• opsworks-instance-assets-eu-ouest-2 • opsworks-instance-assets-eu-west-3 • opsworks-instance-assets-sa-east-1 • opsworks-instance-assets-us-west-1 • opsworks-instance-assets-us-ouest-2	• opsworks-us-west-1 journal • opsworks-us-west-2 journaux	

Pour Chef 11.10 pour Linux et versions antérieures, les compartiments se présentent comme suit. Les piles Chef 11.4 ne sont pas prises en charge sur les points de terminaison régionaux situés en dehors de la région USA Est (Virginie du Nord).

Compartiments d'agents	Compartiments de ressources	Compartiments de journaux	Compartiments DNA
• opsworks-instance-agent-us-est-2 • opsworks-instance-agent-us-east-1 • opsworks-instance-agent-ap-south-1 • opsworks-instance-agent-ap-northeast-1 • opsworks-instance-agent-ap-northeast-2 • opsworks-instance-agent-ap-southeast-1 • opsworks-instance-agent-ap-southeast-2 • opsworks-instance-agent-ca-central-1 • opsworks-instance-agent-eu-central-1 • opsworks-instance-agent-eu-west-1	• opsworks-instance-assets-us-est-2 • opsworks-instance-assets-us-east-1 • opsworks-instance-assets-ap-south-1 • opsworks-instance-assets-ap-northeast-1 • opsworks-instance-assets-ap-northeast-2 • opsworks-instance-assets-ap-southeast-1 • opsworks-instance-assets-ap-southeast-2 • opsworks-instance-assets-ca-central-1 • opsworks-instance-assets-eu-central-1 • opsworks-instance-assets-eu-west-1	• prod_stage-log	• prod_stage-dna

Compartiments d'agents	Compartiments de ressources	Compartiments de journaux	Compartiments DNA
• opsworks-instance-agent-eu-ouest-2 • opsworks-instance-agent-eu-west-3 • opsworks-instance-agent-us-east-1 • opsworks-instance-agent-us-west-1 • opsworks-instance-agent-us-ouest-2	• opsworks-instance-assets-eu-ouest-2 • opsworks-instance-assets-eu-west-3 • opsworks-instance-assets-sa-east-1 • opsworks-instance-assets-us-west-1 • opsworks-instance-assets-us-ouest-2		

Pour plus d'informations, consultez Points de terminaison d'un VPC.

> **Note**
>
> Pour que AWS OpsWorks Stacks puisse se connecter aux points de terminaison VPC que vous activez, vous devez également configurer le routage pour votre NAT ou votre adresse IP publique, car AWS OpsWorks l'agent Stacks a toujours besoin d'accéder au point de terminaison public.

Rubriques

- Notions de base des VPC
- Création d'un VPC pour une pile Stacks AWS OpsWorks

Notions de base des VPC

Pour une discussion détaillée sur VPCs, consultez Amazon Virtual Private Cloud. En bref, un VPC se compose d'un ou plusieurs sous-réseaux, chacun contenant une ou plusieurs instances. Chaque sous-réseau dispose d'une table de routage associée qui dirige le trafic sortant en fonction de son adresse IP de destination.

- Les instances au sein d'un VPC peuvent communiquer les unes avec les autres par défaut, quel que soit leur sous-réseau. Cependant, les modifications apportées aux listes de contrôle d'accès au réseau (ACLs), aux politiques des groupes de sécurité ou à l'utilisation d'adresses IP statiques peuvent interrompre cette communication.
- Les sous-réseaux dont les instances peuvent communiquer avec Internet sont appelés sous-réseaux publics.
- Les sous-réseaux dont les instances peuvent communiquer uniquement avec d'autres instances du VPC et ne peuvent pas communiquer directement avec le réseau Internet sont appelés sous-réseaux privés.

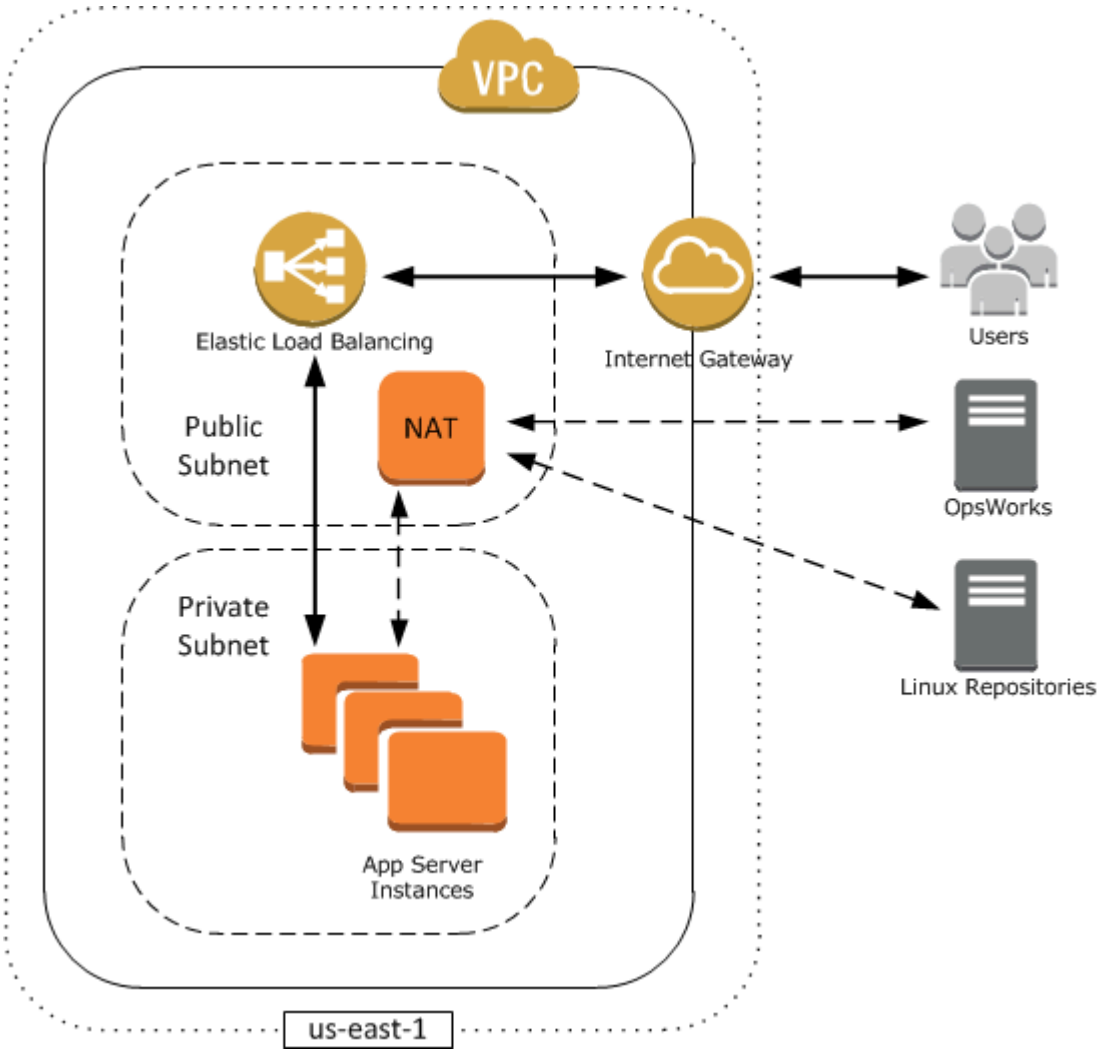
AWS OpsWorks Stacks nécessite que le VPC soit configuré de telle sorte que chaque instance de la pile, y compris les instances situées dans des sous-réseaux privés, ait accès aux points de terminaison suivants :

- L'un des points de terminaison du service AWS OpsWorks Stacks répertoriés dans la section « Support régional » de. Débuter avec AWS OpsWorks Stacks
- L'un des points de terminaison du service d'instance suivants, utilisé par l'agent AWS OpsWorks Stacks. L'agent s'exécute sur les instances gérées par le client pour échanger des données avec le service.
  - 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-serviceeu-west-1.amazonaws.com
- opsworks-instance-serviceeu-west-2.amazonaws.com
- opsworks-instance-serviceeu-west-3.amazonaws.com

- Amazon S3
- Tous les référentiels de package dont votre système d'exploitation dépend, tels que les référentiels Amazon Linux ou Ubuntu Linux.
- Vos référentiels d'applications et de livres de recettes personnalisés.

Il existe différentes façons de configurer un VPC pour assurer cette connectivité. Voici un exemple simple de configuration d'un VPC pour une pile de serveurs d'applications AWS OpsWorks Stacks.



Ce VPC a plusieurs composants :

Sous-réseaux

Le VPC a deux sous-réseaux, l'un public et l'autre privé.

- Le sous-réseau public contient un équilibreur de charge et un périphérique de traduction d'adresses réseau (NAT), qui peuvent communiquer avec des adresses externes et les instances du sous-réseau privé.
- Le sous-réseau privé contient les serveurs d'applications, qui peuvent communiquer avec le périphérique NAT et l'équilibreur de charge du sous-réseau public, mais pas directement avec les adresses externes.

Passerelle Internet

La passerelle Internet permet aux instances ayant des adresses IP publiques, telles que l'équilibreur de charge, de communiquer avec des adresses externes au VPC.

Équilibreur de charge

L'équilibreur de charge Elastic Load Balancing accepte le trafic entrant des utilisateurs, le distribue aux serveurs d'applications du sous-réseau privé et retourne les réponses aux utilisateurs.

NAT

Le périphérique NAT fournit aux serveurs d'applications un accès limité à Internet, généralement utilisé à des fins telles que le téléchargement de mises à jour logicielles à partir d'un référentiel externe. Toutes les instances de AWS OpsWorks Stacks doivent être capables de communiquer avec AWS OpsWorks Stacks et avec les référentiels Linux appropriés. Le seul moyen de gérer ce problème consiste à placer un périphérique NAT avec une adresse IP Elastic associée dans un sous-réseau public. Vous pouvez alors acheminer le trafic sortant à partir des instances du sous-réseau privé via le périphérique NAT.

> **Note**
>
> Une seule instance NAT crée un point de défaillance unique dans le trafic sortant de votre sous-réseau privé. Vous pouvez améliorer la fiabilité en configurant un VPC avec une paire d'instances NAT qui prennent le relais en cas de défaillance de l'une d'entre elles. Pour plus d'informations, consultez Solution haute disponibilité des instances

NAT sur Amazon VPC. Vous pouvez également utiliser une passerelle NAT. Pour plus d'informations, consultez la section NAT dans le guide de l'utilisateur Amazon VPC.

La configuration VPC optimale dépend de votre stack AWS OpsWorks Stacks. Voici quelques exemples où vous pourriez utiliser certaines configurations de VPC. Pour obtenir des exemples de scénarios VPC, consultez Scénarios d'utilisation d'Amazon VPC.

Utilisation d'une instance dans un sous-réseau public

Si vous disposez d'une pile à instance unique sans ressources privées associées, telle qu'une instance Amazon RDS qui ne doit pas être accessible au public, vous pouvez créer un VPC avec un sous-réseau public et placer l'instance dans ce sous-réseau. Si vous n'utilisez pas de VPC par défaut, la couche de l'instance doit assigner une adresse IP Elastic à l'instance. Pour de plus amples informations, veuillez consulter OpsWorks Principes de base des couches.

Utilisation des ressources privées

Si vous avez des ressources qui ne doivent pas être accessibles publiquement, vous pouvez créer un VPC avec un sous-réseau public et un sous-réseau privé. Par exemple, dans un environnement de dimensionnement automatique à équilibrage de charge, vous pouvez placer toutes les EC2 instances Amazon dans le sous-réseau privé et l'équilibreur de charge dans un sous-réseau public. Ainsi, les EC2 instances Amazon ne sont pas directement accessibles depuis Internet ; tout le trafic entrant doit être acheminé via l'équilibreur de charge.

Le sous-réseau privé isole les instances de l'accès EC2 direct des utilisateurs à Amazon, mais elles doivent tout de même envoyer des demandes sortantes à AWS et aux référentiels de packages Linux appropriés. Pour autoriser de telles demandes, vous pouvez, par exemple, utiliser un périphérique de traduction d'adresses réseau (NAT) avec sa propre adresse IP Elastic, puis acheminer le trafic sortant des instances via ce périphérique. Vous pouvez placer le périphérique NAT dans le même sous-réseau public que l'équilibreur de charge, comme illustré dans l'exemple précédent.

- Si vous utilisez une base de données principale telle qu'une instance Amazon RDS, vous pouvez placer ces instances dans le sous-réseau privé. Pour les instances Amazon RDS, vous devez spécifier au moins deux sous-réseaux différents dans des zones de disponibilité différentes.
- Si vous avez besoin d'un accès direct aux instances d'un sous-réseau privé (par exemple, si vous souhaitez utiliser SSH pour vous connecter à une instance), vous pouvez placer un hôte bastion dans le sous-réseau public qui transmet par proxy les demandes provenant d'Internet.

Extension de votre propre réseau dans AWS

Si vous souhaitez étendre votre propre réseau dans le cloud et accéder aussi directement à Internet depuis votre VPC, vous pouvez créer une passerelle VPN. Pour plus d'informations, consultez Scénario 3 : VPC avec sous-réseaux privés et publics, et accès VPN matériel.

Création d'un VPC pour une pile Stacks AWS OpsWorks

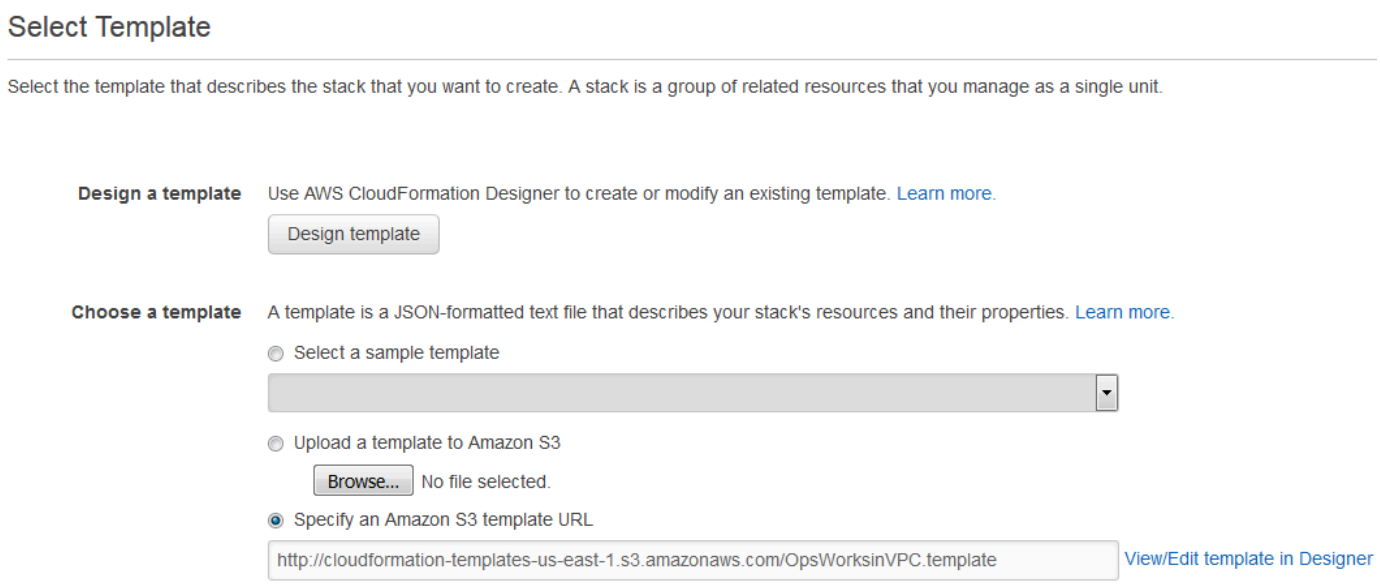
Cette section explique comment créer un VPC pour une pile AWS OpsWorks Stacks à l'aide d'un exemple de modèle AWS. CloudFormation Vous pouvez télécharger le modèle dans le fichier OpsWorks VPCtemplates .zip. Pour plus d'informations sur la création manuelle d'un VPC comme celui présenté dans cette rubrique, consultez Scénario 2 : VPC avec sous-réseaux publics et privés. Pour plus d'informations sur la configuration des tables de routage, des groupes de sécurité, etc., consultez l'exemple de modèle.

> **Note**
>
> Par défaut, AWS OpsWorks Stacks affiche les noms de sous-réseaux en concaténant leur plage d'adresses CIDR et leur zone de disponibilité, par exemple. `10.0.0.1/24 - us-east-1b` Pour rendre les noms plus lisibles, créez une balise pour chaque sous-réseau avec la clé définie sur **`Name`** et la valeur définie sur le nom du sous-réseau. AWS OpsWorks Stacks ajoute ensuite le nom du sous-réseau au nom par défaut. Par exemple, le sous-réseau privé de l'exemple suivant possède une balise dont le nom est défini sur**`Private`**, qui s' OpsWorks affiche sous `10.0.0.1/24 us-east - 1b - Private` la forme.

Vous pouvez lancer un modèle VPC à l'aide de la AWS CloudFormation console en quelques étapes seulement. La procédure suivante utilise l'exemple de modèle pour créer un VPC dans la région US East (N. Virginia). Pour obtenir les instructions sur l'utilisation du modèle pour créer un VPC dans d'autres régions, consultez la Remarque qui suit la procédure.

Pour créer le VPC

1. Ouvrez la console AWS CloudFormation, sélectionnez la région USA Est (Virginie du Nord), puis choisissez Créer une pile.
2. Sur la page Select Template (Sélectionner un modèle), sélectionnez Upload a template (Télécharger un modèle). Recherchez le **`OpsWorksinVPC.template`** fichier que vous avez téléchargé dans le fichier OpsWorks VPCtemplates .zip. Choisissez Continuer.



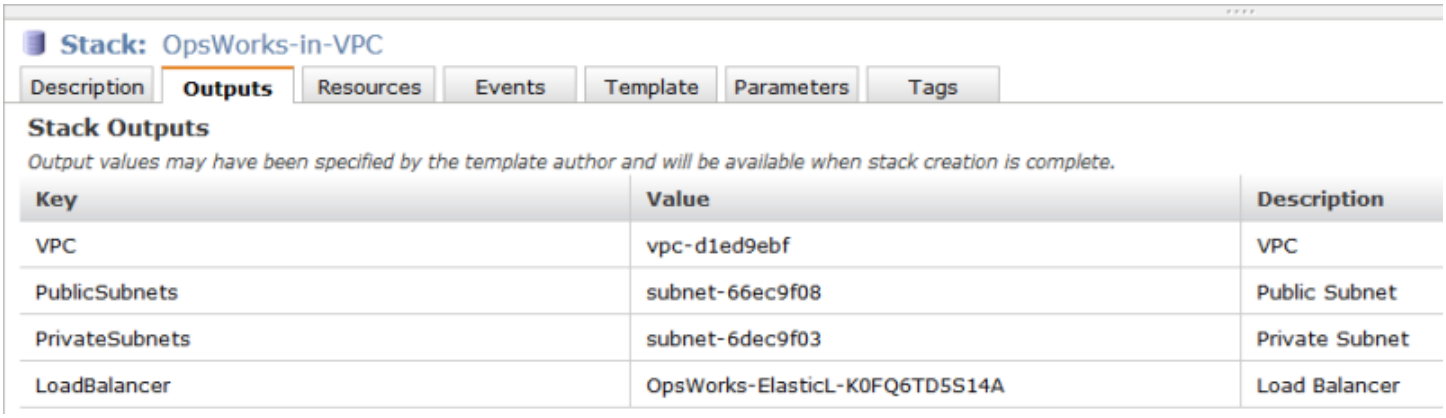
Vous pouvez également lancer cette pile en ouvrant AWS CloudFormation Sample Templates, en localisant le modèle VPC AWS OpsWorks Stacks et en choisissant Launch Stack.

3. Sur la page Spécifier les paramètres, acceptez les valeurs par défaut et choisissez Continuer.
4. Sur la page Ajouter une balise, créez une balise avec Clé défini sur **`Name`** et Valeur défini sur le nom du VPC. Cette balise facilitera l'identification de votre VPC lorsque vous créez une pile AWS OpsWorks Stacks.
5. Choisissez Continuer, puis Fermer pour lancer la pile.

Remarque : Vous pouvez créer le VPC dans d'autres régions en utilisant l'une des approches suivantes.

- Accédez à Utilisation de modèles dans différentes régions, choisissez la région appropriée, recherchez le modèle VPC AWS OpsWorks Stacks, puis choisissez Launch Stack.
- Copiez le fichier modèle sur votre système, sélectionnez la région appropriée dans la AWS CloudFormation console et utilisez l'option Upload a template to Amazon S3 de l'assistant Create Stack pour télécharger le modèle depuis votre système.

L'exemple de modèle inclut des sorties qui fournissent le VPC, le sous-réseau et l'équilibreur de charge IDs dont vous aurez besoin pour créer la pile Stacks. AWS OpsWorks Vous pouvez les voir en choisissant l'onglet Sorties en bas de la fenêtre de AWS CloudFormation console.


Stack: OpsWorks-in-VPC

Description | Outputs | Resources | Events | Template | Parameters | Tags

Stack Outputs

Output values may have been specified by the template author and will be available when stack creation is complete.

Key	Value	Description
VPC	vpc-d1ed9ebf	VPC
PublicSubnets	subnet-66ec9f08	Public Subnet
PrivateSubnets	subnet-6dec9f03	Private Subnet
LoadBalancer	OpsWorks-ElasticL-K0FQ6TD5S14A	Load Balancer

Mise à jour d'une pile

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une fois que vous avez créé une pile, vous pouvez mettre à jour sa configuration à tout moment. Sur la page Stack, cliquez sur Stack Settings, puis cliquez sur Edit pour afficher la page Settings. Apportez les modifications souhaitées et cliquez sur Save.

Les paramètres sont les mêmes que ceux mentionnées dans Créer une pile. Reportez-vous à cette rubrique pour plus de détails. Toutefois, notez les points suivants :

- Vous ne pouvez pas modifier la région ni l'ID du VPC.
- Si votre pile est en cours d'exécution dans un VPC, les paramètres incluent un paramètre Default subnet qui répertorie les sous-réseaux du VPC. Si votre pile n'est pas en cours d'exécution dans un VPC, le paramètre est nommé Default Availability Zones et il répertorie les zones de disponibilité de la région.
- Vous pouvez changer le système d'exploitation par défaut, mais vous ne pouvez pas spécifier de système d'exploitation Linux pour une pile Windows ou de système d'exploitation Windows pour une pile Linux.

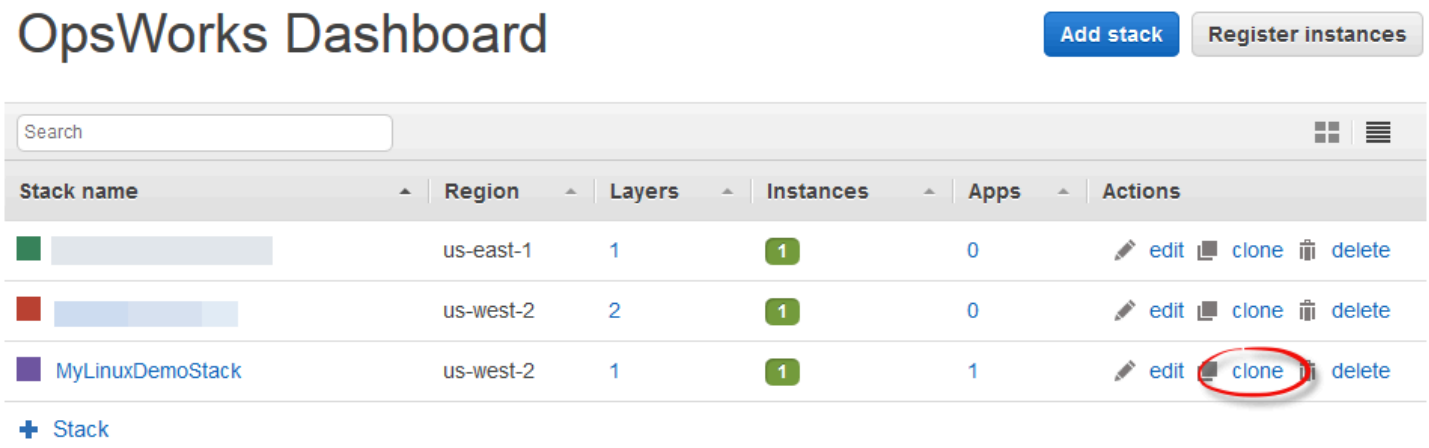
- Si vous modifiez l'un des paramètres par défaut de l'instance, par exemple Hostname theme ou Default SSH key, les nouvelles valeurs s'appliquent uniquement à toutes les nouvelles instances que vous créez, et non aux instances existantes.
- La modification du nom change le nom affiché par la console ; cela ne change pas le nom abrégé sous-jacent utilisé par AWS OpsWorks Stacks pour identifier la pile.
- Avant de passer de Oui à Non à Utiliser les groupes de OpsWorks sécurité, chaque couche doit avoir au moins un groupe de sécurité en plus du groupe de sécurité intégré à la couche. Pour de plus amples informations, veuillez consulter Modification de OpsWorks la configuration d'une couche.

  AWS OpsWorks Stacks supprime ensuite les groupes de sécurité intégrés de chaque couche.
- Si vous remplacez Utiliser les groupes de OpsWorks sécurité par Non par Oui, AWS OpsWorks Stacks ajoute le groupe de sécurité intégré approprié à chaque couche, mais ne supprime pas les groupes de sécurité existants.

Cloner une pile

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Il est parfois utile de créer plusieurs copies d'une pile. Par exemple, vous pouvez ajouter une redondance comme reprise après sinistre ou mesure de prévention, ou vous pouvez utiliser une pile existante comme point de départ pour une nouvelle pile. L'approche la plus simple consiste à cloner la pile source. Sur le tableau de bord AWS OpsWorks Stacks, dans la colonne Actions de la ligne correspondant à la pile que vous souhaitez cloner, choisissez Clone, ce qui ouvre la page Clone stack.



Au départ, les paramètres de la pile clonée sont identiques à ceux de la pile source, sauf que le terme « copy » est ajouté au nom de la pile. Pour plus d'informations sur ces paramètres, consultez Créer une pile. Il existe également deux paramètres facultatifs supplémentaires :

Autorisations

Si Toutes les autorisations est sélectionné (valeur par défaut), les autorisations de pile source sont appliquées à la pile clonée.

Applications

Liste les applications qui sont déployées sur la pile source. Pour chaque application listée, si la case à cocher correspondante est sélectionnée (valeur par défaut), l'application est déployée sur la pile clonée.

> **Note**
>
> Vous ne pouvez pas cloner une pile d'un point de terminaison régional à un autre ; par exemple, vous ne pouvez pas cloner une pile de la région USA Ouest (Oregon) (us-west-2) vers la région Asie-Pacifique (Mumbai) (ap-south-1).

Lorsque vous avez finalisé les paramètres, choisissez Clone stack. AWS OpsWorks Stacks crée une nouvelle pile composée des couches de la pile source et éventuellement de ses applications et autorisations. Les couches ont la même configuration que celles du départ, soumise à des modifications que vous avez faites. Toutefois, le clonage ne crée aucune instance. Vous devez ajouter un ensemble approprié d'instances à chaque couche de la pile clonée, puis les démarrer. Comme avec n'importe quelle pile, vous pouvez exécuter des tâches de gestion standard sur une pile

clonée, par exemple l'ajout, la suppression ou la modification des couches ou l'ajout et le déploiement d'applications.

Pour que la pile clonée soit opérationnelle, démarrez les instances. AWS OpsWorks Stacks installe et configure chaque instance en fonction de son appartenance à la couche. Il déploie également toutes les applications, comme avec une nouvelle pile.

Commandes AWS OpsWorks Run Stacks Stack

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks fournit un ensemble de commandes de pile, que vous pouvez utiliser pour effectuer diverses opérations sur les instances d'une pile. Pour exécuter une commande de pile, cliquez sur Run Command (Exécuter la commande) sur la page Pile. Vous choisissez ensuite la commande appropriée, vous spécifiez n'importe quelle option, puis vous appuyez sur le bouton en bas à droite qui porte le nom de la commande.

> **Note**
>
> AWS OpsWorks Stacks prend également en charge un ensemble de commandes de déploiement, que vous utilisez pour gérer le déploiement des applications. Pour de plus amples informations, veuillez consulter Déploiement d'applications.

Vous pouvez exécuter les commandes de pile suivantes sur n'importe quelle pile.

Update Custom Cookbooks

Met à jour les livres de recettes personnalisés des instances avec la dernière version du référentiel. Cette commande n'exécute aucune recette. Pour exécuter les recettes mises à jour, vous pouvez utiliser une commande de pile `Execute Recipes`, `Setup` ou `Configure`, ou

redéployer votre application pour exécuter les recettes de déploiement. Pour plus d'informations sur les livres de recettes personnalisés, consultez Livres de recettes et recettes.

Execute Recipes

Exécute un ensemble de recettes sur les instances. Pour de plus amples informations, veuillez consulter Exécution manuelle des recettes.

Configuration

Exécute les recettes de configuration des instances.

Configuration

Exécute des recettes de configuration des instances.

> **Note**
>
> Pour que vous puissiez utiliser Setup ou Configure afin d'exécuter des recettes sur une instance, les recettes doivent être attribuées à l'événement de cycle de vie correspondant pour la couche de l'instance. Pour de plus amples informations, veuillez consulter Exécution des recettes.

Vous pouvez exécuter les commandes de pile suivantes uniquement sur les piles Linux.

Install Dependencies

Installe les packages des instances. A partir de Chef 12, cette commande n'est pas disponible.

Update Dependencies

(Linux uniquement. À partir de Chef 12, cette commande n'est plus disponible.) Installe les mises à jour habituelles des systèmes d'exploitation et des packages. Les détails dépendent du système d'exploitation des instances. Pour de plus amples informations, veuillez consulter Gestion des mises à jour de sécurité.

Utilisez la commande Mettre à niveau le système d'exploitation pour mettre à niveau les instances vers une nouvelle version d'Amazon Linux .

Upgrade Operating System

(Linux uniquement) Met à niveau les systèmes d'exploitation Amazon Linux des instances vers la dernière version. Pour de plus amples informations, veuillez consulter AWS OpsWorks Systèmes d'exploitation Stacks.

> **Important**
>
> Une fois que vous avez exécuté Mettre à niveau le système d'exploitation, nous vous recommandons d'exécuter aussi Configuration. Vous pouvez ainsi vous assurer que les services sont redémarrés correctement.

Les commandes de pile ont les options suivantes, dont certaines apparaissent uniquement pour certaines commandes.

Comment

(Facultatif) Entrez des remarques personnalisées.

Recipes to execute

(Obligatoire) Ce paramètre s'affiche uniquement si vous sélectionnez la commande Exécuter les recettes. Entrez les recettes à exécuter en utilisant le *recipe_name* format standard *cookbook_name* : :, séparées par des virgules. Si vous spécifiez plusieurs recettes, AWS OpsWorks Stacks les exécute dans l'ordre indiqué.

Allow reboot

(Facultatif) Ce paramètre s'affiche uniquement si vous sélectionnez la commande Upgrade Operating System. La valeur par défaut est Yes, ce qui indique à AWS OpsWorks Stacks de redémarrer les instances après avoir installé la mise à niveau.

JSON Chef personnalisé

(Facultatif) Choisissez Avancé pour afficher cette option qui vous permet de spécifier des attributs personnalisés JSON devant être intégrés aux attributs de configuration et de déploiement de pile.

instances

(Facultatif) Spécifiez les instances sur lesquelles exécuter la commande. Toutes les instances en ligne sont sélectionnées par défaut. Pour exécuter la commande sur un sous-ensemble d'instances, sélectionnez les couches ou instances appropriées.

> **Note**
>
> Vous pouvez voir les exécutions d'execute_recipes que vous n'avez pas exécutées sur les pages Déploiement et Commandes. C'est généralement dû à un changement d'autorisation, par exemple l'octroi ou la suppression des autorisations SSH pour un utilisateur. Lorsque vous apportez une telle modification, AWS OpsWorks Stacks utilise execute_recipes pour mettre à jour les autorisations sur les instances.

Utilisation du JSON personnalisé

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Plusieurs actions AWS OpsWorks Stacks vous permettent de spécifier un JSON personnalisé, que AWS OpsWorks Stacks installe sur les instances et peut être utilisé par des recettes.

Vous pouvez spécifier un JSON personnalisé dans les cas suivants :

- Lorsque vous créez, mettez à jour ou clonez une pile.

  AWS OpsWorks Stacks installe le JSON personnalisé sur toutes les instances pour tous les événements du cycle de vie ultérieurs.
- Lorsque vous exécutez une commande déploiement ou de pile.

  AWS OpsWorks Stacks transmet le JSON personnalisé aux instances uniquement pour cet événement.

Le JSON personnalisé doit être représenté par, et formaté comme, un objet JSON valide. Par exemple :

```
{
  "att1": "value1",
  "att2": "value2"
  ...
}
```

AWS OpsWorks Stacks stocke du code JSON personnalisé aux emplacements suivants :

Sur les instances Linux :

- `/var/chef/runs/`*`run-ID`*`/attribs.json`
- `/var/chef/runs/`*`run-ID`*`/nodes/`*`hostname`*`.json`

Sur les instances Windows :

- *`drive`*`:\chef\runs\`*`run-ID`*`\attribs.json`
- *`drive`*`:\chef\runs\`*`run-ID`*`\nodes\`*`hostname`*`.json`

> **Note**
>
> Dans Chef 11.10 et les versions antérieures pour Linux, le JSON personnalisé se trouve dans le chemin d'accès suivant sur les instances Linux, les instances Windows ne sont pas disponibles et il n'y a aucun fichier `attribs.json`. Les journaux sont stockes dans le même répertoire ou dossier que le JSON. Pour plus d'informations sur le JSON personnalisé dans Chef 11.10 et les versions antérieures pour Linux, consultez Remplacement des attributs par le JSON personnalisé et Journaux de Chef.
> `/var/lib/aws/opsworks/chef/`*`hostname`*`.json`

Dans les chemins précédents, *`run-ID`* il y a un identifiant unique que AWS OpsWorks Stacks attribue à chaque exécution de Chef sur une instance, et *`hostname`* c'est le nom d'hôte de l'instance.

Pour accéder au JSON personnalisé à partir des recettes de Chef, utilisez la syntaxe `node` standard de Chef.

Par exemple, supposons que vous souhaitiez définir des paramètres simples pour une application que vous souhaitez déployer, par exemple si l'application est visible au départ et les couleurs initiales

de premier plan et d'arrière plan de l'application. Supposons que vous définissez ces paramètres d'application avec un objet JSON comme suit :

```
{
  "state": "visible",
  "colors": {
    "foreground": "light-blue",
    "background": "dark-gray"
  }
}
```

Pour déclarer le JSON personnalisé pour une pile :

1. Sur la page de la pile, choisissez Paramètres de pile, puis Modifier.
2. Pour Custom Chef JSON (Chef JSON personnalisé), saisissez l'objet JSON et choisissez Enregistrer.

> **Note**
>
> Vous pouvez déclarer un JSON personnalisé aux niveaux du déploiement, de la couche et de la pile. Vous pouvez le faire si vous voulez qu'une partie du JSON personnalisé soit uniquement visible par un déploiement et une couche. Vous pouvez aussi chercher à remplacer temporairement un JSON personnalisé déclaré au niveau de la pile par un JSON personnalisé déclaré au niveau de la couche. Si vous déclarez un JSON personnalisé à plusieurs niveaux, celui qui est déclaré au niveau du déploiement remplace n'importe quel JSON personnalisé déclaré aux niveaux de la couche et de la pile. Le JSON personnalisé déclaré au niveau de la couche remplace n'importe quel JSON personnalisé déclaré uniquement au niveau de la pile.
>
> Pour utiliser la console AWS OpsWorks Stacks afin de spécifier le JSON personnalisé pour un déploiement, sur la page Déployer l'application, sélectionnez Avancé. Tapez le JSON personnalisé dans la zone Custom Chef JSON (JSON Chef personnalisé), puis choisissez Enregistrer.
>
> Pour utiliser la console AWS OpsWorks Stacks afin de spécifier un code JSON personnalisé pour une couche, sur la page Couches, sélectionnez Paramètres pour la couche souhaitée. Saisissez le JSON personnalisé dans la zone JSON personnalisé, puis choisissez Enregistrer.

Pour plus d'informations, consultez Modification de OpsWorks la configuration d'une couche et Déploiement d'applications.

Lorsque vous exécutez une commande déploiement ou de pile, les recettes peuvent extraire ces valeurs personnalisées à l'aide de la syntaxe node standard de Chef, qui mappe directement à la hiérarchie de l'objet JSON personnalisé. Par exemple, le code de recette suivant écrit des messages dans le journal de Chef sur les valeurs JSON personnalisées précédentes :

```
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']}' **********")
```

Cette approche peut être utile pour transmettre des données aux recettes. AWS OpsWorks Stacks ajoute ces données à l'instance, et les recettes peuvent récupérer les données en utilisant la node syntaxe standard de Chef.

Note

Le JSON personnalisé est limité à 120 Ko. Si vous avez besoin de plus de capacité, nous vous recommandons de stocker certaines données sur Amazon Simple Storage Service (Amazon S3). Vos recettes personnalisées peuvent ensuite utiliser l'interface de ligne de commande AWS ou le AWS SDK pour Rubypour télécharger les données du compartiment Amazon S3 vers votre instance.

Suppression d'une pile

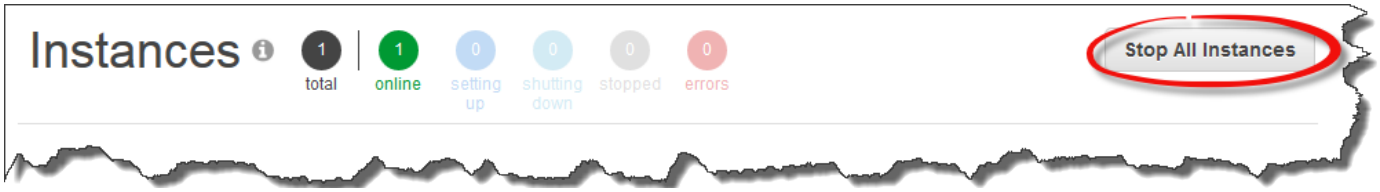
Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Si vous n'avez plus besoin d'une pile, vous pouvez la supprimer. Seuls des piles vides peuvent être supprimées. Vous devez d'abord supprimer toutes les instances, les applications et les couches figurant dans la pile.

Pour supprimer une pile

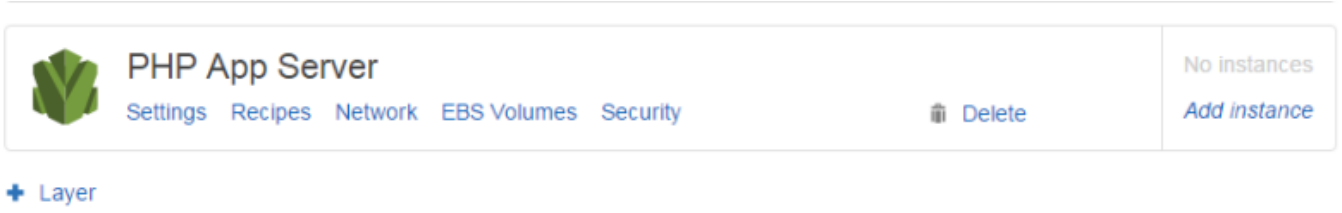
1. Dans le tableau de bord AWS OpsWorks Stacks, sélectionnez la pile que vous souhaitez fermer et supprimer.
2. Dans le panneau de navigation, choisissez Instances.
3. Dans la page Instances, choisissez Stop all Instances (Arrêter toutes les instances).



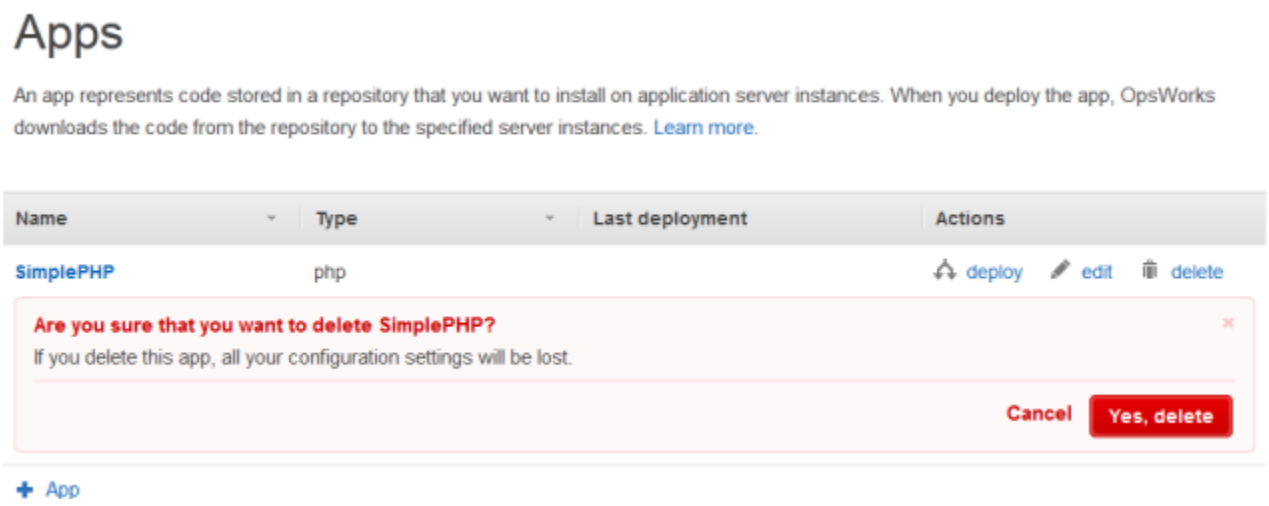
4. Une fois que les instances ont été arrêtées, pour chaque instance de la couche, choisissez Supprimer dans la colonne Actions. Lorsque vous êtes invité à confirmer, choisissez Oui, supprimer.



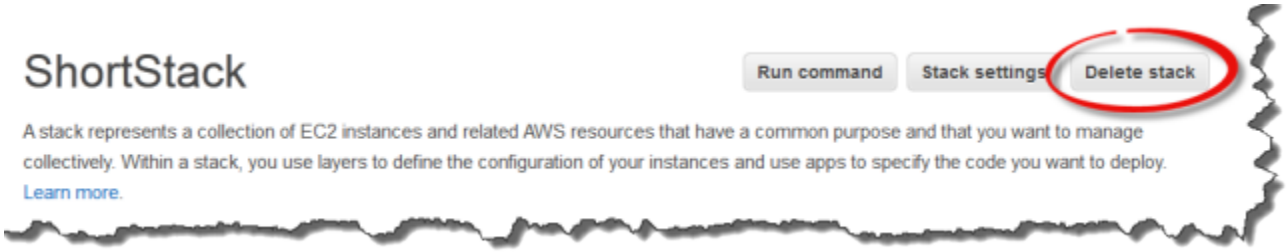
5. Lorsque toutes les instances ont été supprimées, dans le volet de navigation, choisissez Layers (Couches).
6. Dans la page Layers (Couches), pour chaque couche de la pile, choisissez Supprimer. Lorsque vous êtes invité à confirmer l'opération, choisissez Oui, supprimer.



7. Lorsque toutes les couches ont été supprimées, dans le volet de navigation, choisissez Apps (Applications).
8. Dans la page Apps (Applications), pour chaque application de la pile, choisissez Supprimer dans la colonne Actions. Lorsque vous êtes invité à confirmer l'opération, choisissez Oui, supprimer.



9. Lorsque toutes les applications ont été supprimées, dans le volet de navigation, choisissez Stack (Pile).
10. Dans la page de la pile, choisissez Delete stack (Supprimer la pile). Lorsque vous êtes invité à confirmer l'opération, choisissez Oui, supprimer.



Supprimer d'autres AWS ressources utilisées par une pile

Vous utilisez d'autres AWS ressources avec AWS OpsWorks Stacks pour créer et gérer vos piles. Lorsque vous supprimez une pile, pensez également à supprimer les ressources qui ont fonctionné

avec la pile, si aucune autre pile ne les utilise et si des ressources extérieures à la AWS OpsWorks pile ne les utilisent pas. Vous trouverez ci-dessous des suggestions de raisons pour nettoyer les AWS ressources externes que vous avez utilisées avec une pile.

- Les AWS ressources externes peuvent continuer à être facturées sur votre AWS compte.
- Les ressources telles que les compartiments Amazon S3 peuvent contenir des informations d'identification personnelle, sensibles ou confidentielles.

> **Important**
>
> Ne supprimez pas ces ressources si elles sont utilisées par d'autres piles. Notez que les rôles IAM et les groupes de sécurité sont globaux, de sorte que les piles d'autres régions peuvent utiliser ces mêmes ressources.

Vous trouverez ci-dessous d'autres AWS ressources utilisées par Stacks, ainsi que des liens vers des informations sur la manière de les supprimer.

Rôles de service et profils d'instance

Lorsque vous créez une pile, vous spécifiez un rôle IAM et un profil d'instance que AWS OpsWorks Stacks utilise pour créer des ressources autorisées en votre nom. AWS OpsWorks crée le profil de rôle et d'instance pour vous si vous ne choisissez aucun profil existant. Le rôle et le profil d'instance AWS OpsWorks créés pour vous sont respectivement nommés `aws-opsworks-service-role` et`aws-opsworks-ec2-role`. Si aucune autre pile de votre compte n'utilise le rôle IAM et le profil d'instance, vous pouvez supprimer ces ressources en toute sécurité. Pour plus d'informations sur la suppression de rôles et de profils d'instance IAM, consultez la page Suppression de rôles ou de profils d'instance.

Groupes de sécurité

Dans AWS OpsWorks Stacks, vous pouvez spécifier des groupes de sécurité définis par l'utilisateur au niveau de la couche. Vous créez des groupes de sécurité à l'aide de la EC2 console ou de l'API Amazon. Des piles et des couches figurant dans d'autres régions peuvent utiliser les mêmes groupes de sécurité, car les groupes de sécurité sont globaux. Vous pouvez supprimer un groupe de sécurité s'il n'est pas utilisé par d'autres AWS ressources. Pour plus d'informations sur la suppression d'un groupe de sécurité, consultez Supprimer un groupe de sécurité dans le guide de EC2 l'utilisateur Amazon.

Volumes Amazon EBS

Dans AWS OpsWorks Stacks, vous ajoutez des volumes EBS au niveau de la couche, et ils sont attachés aux instances de la couche. Vous créez des volumes EBS à l'aide de la console de EC2 service ou de l'API Amazon, puis vous les attachez aux instances AWS OpsWorks Stacks au niveau de la couche. Les volumes EBS sont spécifiques à une zone de disponibilité. Si vous n'utilisez plus un volume EBS dans aucune pile d'une zone de disponibilité et d'une région spécifique, vous pouvez supprimer ce volume. Pour plus d'informations sur la suppression d'un volume Amazon EBS, consultez Supprimer un volume Amazon EBS dans le guide de EC2 l'utilisateur Amazon.

Compartiments Amazon Simple Storage Service (Amazon S3)

Dans AWS OpsWorks Stacks, vous pouvez utiliser les compartiments Amazon S3 pour les opérations suivantes. Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

- Stockage d'un code d'application
- Stockage de recettes et de livres de recettes
- CloudTrail journaux, si vous avez activé la CloudTrail connexion à AWS OpsWorks Stacks
- Streams Amazon CloudWatch Logs, si vous les avez activés dans AWS OpsWorks Stacks

Adresses IP Elastic

Si vous avez enregistré des adresses IP Elastic auprès de AWS OpsWorks Stacks et que vous n'en avez plus besoin, vous pouvez libérer l'adresse IP Elastic.

Equilibreurs de charge Elastic Load Balancing

Si vous n'avez plus besoin d'un équilibreur de charge classique Elastic Load Balancing que vous utilisiez avec des couches dans votre pile, vous pouvez le supprimer. Pour plus d'informations, consultez Suppression de votre équilibreur de charge dans le Guide de l'utilisateur pour les équilibreurs de charge classiques.

Instances Amazon Relational Database Service (Amazon RDS)

Si vous avez enregistré des instances de base de données (DB) Amazon RDS auprès de AWS OpsWorks Stacks et que vous n'en avez plus besoin, vous pouvez supprimer des instances de base de données. Pour plus d'informations sur la façon de supprimer des instances de base de

données, consultez Supprimer une instance de base de données dans le guide de l'utilisateur Amazon RDS.

Clusters Amazon Elastic Container Service (Amazon ECS)

Si votre pile comprenait des couches de cluster ECS et que vous n'utilisez plus le cluster ECS qui a été enregistré avec une couche, vous pouvez supprimer ce cluster ECS. Pour plus d'informations sur la suppression d'un cluster ECS, consultez Supprimer un cluster dans le manuel Amazon ECS Developer Guide.

Couches

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Chaque pile contient une ou plusieurs couches, chacune représentant un composant de la pile, telle qu'un équilibreur de charge ou un ensemble de serveurs d'applications.

Lorsque vous utilisez des couches AWS OpsWorks Stacks, ayez les points suivants à l'esprit :

- Chaque couche d'une pile doit avoir au moins une instance et peut le cas échéant comporter plusieurs instances.
- Chaque instance d'une pile doit être membre d'au moins une couche, à l'exception des instances enregistrées.

  Vous ne pouvez pas configurer une instance directement, à l'exception de certains paramètres de base tels que la clé SSH et le nom d'hôte. Vous devez créer et configurer une couche appropriée, puis ajouter l'instance à la couche.

EC2 Les instances Amazon peuvent éventuellement être membres de plusieurs couches. Dans ce cas, AWS OpsWorks Stacks exécute les recettes pour installer et configurer des packages, déployer des applications, etc. pour chacune des couches de l'instance.

En attribuant une instance à plusieurs couches, vous pouvez, par exemple, bénéficier des avantages suivants :

- Réduire les dépenses en hébergeant le serveur de base de données et l'équilibreur de charge sur une seule instance.
- Utilisez l'un de vos serveurs d'applications pour l'administration.

  Créez une couche d'administration personnalisée et ajoutez l'une des instances serveur d'applications à cette couche. Les recettes de la couche d'administration configurent l'instance serveur d'applications pour qu'elle effectue les tâches administratives, et installent les logiciels supplémentaires requis. Les autres instances serveur d'applications sont juste des serveurs d'applications.

Cette section décrit comment utiliser les couches.

Rubriques

- OpsWorks Principes de base des couches
- Couche Elastic Load Balancing
- Couche de service Amazon RDS
- Couches de cluster ECS
- Couches AWS OpsWorks de piles personnalisées
- Installations du package de système d'exploitation par couche

OpsWorks Principes de base des couches

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section décrit comment effectuer des opérations communes à toutes les couches AWS OpsWorks Stacks.

Rubriques

- Création d'une OpsWorks couche
- Modification de OpsWorks la configuration d'une couche
- Utilisation de la réparation automatique pour remplacer les instances en échec
- Supprimer une OpsWorks couche

Création d'une OpsWorks couche

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Lorsque vous créez une pile, vous voyez la page suivante :



Pour ajouter la première OpsWorks couche

1. Cliquez sur Add a layer.
2. Sur la page Add Layer, sélectionnez la couche appropriée, ce qui affiche les options de configuration de la couche.
3. Configurez la couche de façon appropriée, puis cliquez sur Add Layer pour l'ajouter à la pile. Les sections suivantes décrivent la configuration des différentes couches.

> **Note**
>
> La page Add Layer affiche uniquement les paramètres de configuration les plus couramment utilisés pour chaque couche. Vous pouvez spécifier des paramètres supplémentaires en procédant à la modification de la couche.

4. Ajoutez les instances à la couche et démarrez-les.

> **Note**
>
> Si une instance est membre de plusieurs couches, vous devez l'ajouter à chacune d'elles avant de lancer l'instance. Vous ne pouvez pas ajouter d'instance en ligne à une couche.

Pour ajouter d'autres couches, ouvrez la page Layers et cliquez sur + Layer pour ouvrir la page Add Layer.

Lorsque vous démarrez une instance, AWS OpsWorks Stacks exécute automatiquement les recettes de configuration et de déploiement pour chacune des couches de l'instance afin d'installer et de configurer les packages appropriés et de déployer les applications appropriées. Vous pouvez personnaliser la configuration et le processus de configuration d'une couche de différentes manières, par exemple en attribuant des recettes personnalisées aux événements du cycle de vie appropriés. AWS OpsWorks Stacks propose des recettes personnalisées d'après les recettes standard pour chaque événement. Pour de plus amples informations, veuillez consulter Livres de recettes et recettes.

Les sections spécifiques aux couches suivantes décrivent comment gérer les étapes 2 et 3 pour les différentes couches AWS OpsWorks Stacks. Pour obtenir plus d'informations sur l'ajout des instances, consultez Ajout d'une instance à une couche.

Modification de OpsWorks la configuration d'une couche

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une fois que vous avez créé une couche, certaines propriétés (par exemple, la région AWS) sont immuables, mais vous pouvez modifier la plus grande partie de la configuration de la couche à tout moment. La modification de la couche permet également d'accéder aux paramètres de configuration qui ne sont pas disponibles sur la page Add Layer. Les paramètres prennent effet dès que vous enregistrez la nouvelle configuration.

Pour modifier une OpsWorks couche

1. Dans le volet de navigation, cliquez sur Layers.
2. Sur la page Layers, choisissez le nom d'une couche pour ouvrir la page des détails qui affiche la configuration actuelle.

   > **Note**
   >
   > Choisissez l'un des noms sous le nom de la couche pour accéder directement à l'onglet associé sur la page des détails.

3. Cliquez sur Edit et sélectionnez l'onglet approprié : General Settings, Recipes, Network, EBS Volumes ou Security.

Les sections suivantes décrivent les paramètres des différents onglets qui sont disponibles pour toutes les couches. Certaines couches ont des paramètres supplémentaires qui leur sont propres et apparaissent en haut de la page. De plus, certains paramètres sont disponibles uniquement pour les piles basées sur Linux, comme indiqué.

Rubriques

- General Settings
- Recettes
- Réseau
- Volumes EBS
- Sécurité
- CloudWatch Journaux

- Balises

General Settings

Toutes les couches ont les paramètres suivants :

Auto healing enabled

Indique si la réparation automatique est activée pour les instances de la couche. Le paramètre par défaut est Yes.

JSON personnalisé

Données au format JSON qui sont transmises aux recettes Chef pour toutes les instances de cette couche. Vous pouvez notamment utiliser ce paramètre pour transmettre les données à vos propres recettes. Pour de plus amples informations, veuillez consulter Utilisation du JSON personnalisé.

> **Note**
>
> Vous pouvez déclarer un JSON personnalisé aux niveaux du déploiement, de la couche et de la pile. Ce paramètre est conseillé si vous souhaitez qu'un JSON personnalisé soit visible sur la pile ou uniquement dans le cadre d'un déploiement individuel. Vous pouvez aussi chercher à remplacer temporairement un JSON personnalisé déclaré au niveau de la couche par un JSON personnalisé déclaré au niveau du déploiement. Si vous déclarez un JSON personnalisé à plusieurs niveaux, celui qui est déclaré au niveau du déploiement remplace n'importe quel JSON personnalisé déclaré aux niveaux de la couche et de la pile. Le JSON personnalisé déclaré au niveau de la couche remplace n'importe quel JSON personnalisé déclaré uniquement au niveau de la pile.
> Pour utiliser la console AWS OpsWorks Stacks afin de spécifier le JSON personnalisé pour un déploiement, sur la page Déployer l'application, sélectionnez Avancé. Tapez le JSON personnalisé dans la zone Custom Chef JSON (JSON Chef personnalisé), puis choisissez Enregistrer.
> Pour utiliser la console AWS OpsWorks Stacks afin de spécifier le JSON personnalisé pour une pile, sur la page des paramètres de la pile, tapez le JSON personnalisé dans la zone JSON personnalisé, puis choisissez Enregistrer.
> Pour plus d'informations, consultez Utilisation du JSON personnalisé et Déploiement d'applications.

Instance shutdown timeout

Spécifie le temps (en secondes) pendant lequel AWS OpsWorks Stacks attend après le déclenchement d'un événement du cycle de vie d'arrêt avant d'arrêter ou de mettre fin à l'instance Amazon. EC2 Le paramètre par défaut est de 120 secondes. L'objectif de ce paramètre est de donner suffisamment de temps aux recettes Shutdown de l'instance pour effectuer leurs tâches avant de mettre hors service l'instance. Si vous pensez que vos recettes Shutdown personnalisées ont besoin de plus de temps, modifiez le paramètre en conséquence. Pour plus d'informations sur l'arrêt des instances, consultez Arrêt d'une instance.

Les autres paramètres de cet onglet varient selon le type de couche et sont identiques aux paramètres de la page Add Layer (Ajouter une couche) de la couche.

Recettes

L'onglet Recettes inclut les paramètres suivants.

Recettes Chef personnalisées

Vous pouvez attribuer les recettes Chef personnalisées à des événements de cycle de vie de la couche. Pour de plus amples informations, veuillez consulter Exécution des recettes.

Réseau

L'onglet Réseau inclut les paramètres suivants.

Elastic Load Balancing

Vous pouvez attacher un équilibreur de charge Elastic Load Balancing à n'importe quelle couche. AWS OpsWorks Stacks enregistre ensuite automatiquement les instances en ligne de la couche auprès de l'équilibreur de charge et les annule lorsqu'elles sont hors ligne. Si vous avez activé la fonction de vidange des connexions de l'équilibreur de charge, vous pouvez spécifier si AWS OpsWorks Stacks la prend en charge. Pour de plus amples informations, veuillez consulter Couche Elastic Load Balancing.

Attribuer automatiquement les adresses IP

Vous pouvez contrôler si AWS OpsWorks Stacks attribue automatiquement des adresses IP publiques ou élastiques aux instances de la couche. Voici ce qui se produit lorsque vous activez cette option :

- Pour les instances sauvegardées par exemple, AWS OpsWorks Stacks attribue automatiquement une adresse chaque fois que l'instance est démarrée.
- Pour les instances basées sur Amazon EBS, AWS OpsWorks Stacks attribue automatiquement une adresse lorsque l'instance est démarrée pour la première fois.
- Si une instance appartient à plusieurs couches, AWS OpsWorks Stacks attribue automatiquement une adresse si vous avez activé l'attribution automatique pour au moins une des couches,

> **Note**
>
> Si vous activez l'attribution automatique d'adresses IP publiques, cela ne s'applique qu'aux nouvelles instances. AWS OpsWorks Stacks ne peut pas mettre à jour l'adresse IP publique des instances existantes.

Si votre pile est en cours d'exécution dans un VPC, vous avez des paramètres distincts pour les adresses publiques et IP Elastic. Le tableau suivant explique comment elles interagissent :

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**
>
> Les instances doivent disposer d'un moyen de communiquer avec le service AWS OpsWorks Stacks, les référentiels de packages Linux et les référentiels de livres de recettes. Si vous ne spécifiez aucune adresse publique ou IP Elastic, votre VPC doit inclure un composant tel qu'un périphérique NAT qui permet aux instances de la couche de communiquer avec des sites externes. Pour de plus amples informations, veuillez consulter Running a Stack in a VPC.

Si votre pile n'est pas exécutée dans un VPC, Adresses IP Elastic est votre seul paramètre :

- Oui : les instances reçoivent une adresse IP Elastic lorsqu'elles sont démarrées pour la première fois ou une adresse IP publique si une adresse IP Elastic ne peut pas être attribuée.
- Non : les instances reçoivent une adresse IP publique chaque fois qu'elles sont démarrées.

Volumes EBS

L'onglet Volumes EBS inclut les paramètres suivants.

EBS optimized instances

Si les instances de la couche doivent être optimisées pour Amazon Elastic Block Store (Amazon EBS). Pour plus d'informations, consultez Instances optimisées pour Amazon EBS.

Volumes EBS supplémentaires

(Linux uniquement) Vous pouvez ajouter des volumes Amazon EBS aux instances de la couche ou les supprimer. Lorsque vous démarrez une instance, AWS OpsWorks Stacks crée automatiquement les volumes et les attache aux instances. Vous pouvez utiliser la page Ressources pour gérer les volumes EBS d'une pile. Pour de plus amples informations, veuillez consulter Gestion des ressources.

- Point de montage — (Obligatoire) Spécifiez le point de montage ou le répertoire dans lequel le volume EBS sera monté.
- # Disques — (Facultatif) Si vous avez spécifié une matrice RAID, le nombre de disques de la matrice.

  Chaque niveau RAID a un nombre de disques par défaut, mais vous pouvez sélectionner un plus grand nombre dans la liste.
- Taille totale (GiB) — (Obligatoire) Taille du volume, en GiB.

  Pour une grappe RAID, ce paramètre spécifie la taille totale de la grappe, et non la taille de chaque disque.

  Le tableau ci-après indique les tailles maximale et minimale autorisées pour chaque type de volume.

Type de volume	Taille minimale (Gio)	Taille maximale (Gio)
Magnétique	1	1 024

Type de volume	Taille minimale (Gio)	Taille maximale (Gio)
IOPS provisionnés (SSD)	4	16384
Usage général (SSD)	1	16384
Débit optimisé (HDD)	500	16384
HDD à froid	500	16384

- Type de volume — (facultatif) : indiquez si vous souhaitez créer un SSD magnétique à usage général, un HDD optimisé pour le débit, un HDD à froid ou un volume PIOPS.

  La valeur par défaut est Magnétique.
- Chiffré — (Facultatif) Spécifiez s'il faut chiffrer le contenu du volume EBS.
- IOPS par disque — (obligatoire pour les volumes SSD IOPS provisionnés et les volumes SSD à usage général) Si vous spécifiez un SSD IOPS provisionné ou un volume SSD à usage général, vous devez également spécifier le nombre d'IOPS par disque.

  Pour des volumes IOPS provisionnés, vous pouvez spécifier la vitesse des opérations d'E/S par seconde lorsque vous créez le volume. Le rapport maximum des opérations d'E/S par seconde provisionnées sur la taille du volume requis peut être de 30 (en d'autres termes, un volume avec 3 000 opérations d'E/S par seconde doit au moins faire 100 Go). Les types de volumes à usage général (SSD) ont un point de départ IOPS de la taille du volume x 3 avec un maximum de 10 000 IOPS et pouvant émettre en rafales jusqu'à 3 000 IOPS pendant 30 minutes.

Lorsque vous ajoutez des volumes sur une couche ou que vous en supprimez, notez les éléments suivants :

- Si vous ajoutez un volume, chaque nouvelle instance obtient le nouveau volume, mais AWS OpsWorks Stacks ne met pas à jour les instances existantes.
- Si vous supprimez un volume, cela s'applique uniquement aux nouvelles instances ; les instances existantes conservent leurs volumes.

Spécification d'un point de montage

Vous pouvez spécifier n'importe quel point de montage à votre convenance. Sachez toutefois que certains points de montage sont réservés à AWS OpsWorks Stacks ou Amazon EC2 et ne doivent

pas être utilisés pour les volumes Amazon EBS. N'utilisez pas les dossiers système Linux typiques tels que `/home` ou `/etc`.

Les points de montage suivants sont réservés à l'usage de AWS OpsWorks Stacks.

- `/srv/www`
- `/var/log/apache2` (Ubuntu)
- `/var/log/httpd` (Amazon Linux)
- `/var/log/mysql`
- `/var/www`

Lorsqu'une instance démarre ou redémarre, autofs (un processus de montage automatique) utilise des points de montage de périphérique éphémères comme `/media/ephemeral0` pour les montages liés. Cette opération a lieu avant le montage des volumes Amazon EBS. Pour vous assurer que le point de montage de votre volume Amazon EBS n'entre pas en conflit avec les fichiers autofs, ne spécifiez pas de point de montage de périphérique éphémère. Les points de montage des appareils éphémères possibles dépendent du type d'instance concerné et du fait qu'il s'agisse d'une instance sauvegardée par un magasin ou d'Amazon EBS. Afin d'éviter tout conflit avec autofs, procédez comme suit :

- Vérifiez les points de montage d'appareils éphémères pour le type d'instance et de stockage que vous souhaitez utiliser.
- Sachez qu'un point de montage qui fonctionne pour une instance basée sur un stockage d'instance peut entrer en conflit avec autofs si vous passez à une instance basée sur Amazon EBS, ou vice versa.

> **Note**
>
> Si vous voulez modifier le mappage de périphérique de stockage en mode bloc du stockage d'instance, vous pouvez créer une AMI personnalisée. Pour plus d'informations, consultez Amazon EC2 Instance Store. Pour plus d'informations sur la création d'une AMI personnalisée pour AWS OpsWorks Stacks, consultezUtilisation de Custom AMIs.

Voici un exemple de l'utilisation d'une recette personnalisée afin de veiller à ce que le point de montage d'un volume ne soit pas en conflit avec autofs. Vous pouvez l'adapter en fonction des besoins de votre cas d'utilisation.

Pour éviter un point de montage en conflit

1. Attribuez un volume Amazon EBS à la couche souhaitée, mais utilisez un point de montage tel `/mnt/workspace` que celui-ci n'entrera jamais en conflit avec les fichiers autofs.
2. Implémentez la recette personnalisée suivante, qui crée un répertoire d'applications sur le volume Amazon EBS et crée des liens vers celui-ci à partir de `/srv/www/` celui-ci. Pour plus d'informations sur l'implémentation des recettes personnalisées, consultez Livres de recettes et recettes et Personnalisation des piles 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. Ajoutez une ligne `depends 'deploy'` au fichier `metadata.rb` du livre de recettes personnalisé.
4. Attribuez cette recette à l'événement Setup de la couche.

Sécurité

L'onglet Security inclut les paramètres suivants.

Groupes de sécurité

Une couche doit être associée à au moins un groupe de sécurité. Vous spécifiez comment associer les groupes de sécurité lorsque vous créez ou mettez à jour une pile. AWS OpsWorks Stacks fournit un ensemble standard de groupes de sécurité intégrés.

- L'option par défaut consiste à faire en AWS OpsWorks sorte que Stacks associe automatiquement le groupe de sécurité intégré approprié à chaque couche.
- Vous pouvez également choisir de ne pas associer automatiquement les groupes de sécurité intégrés, mais d'associer un groupe de sécurité personnalisé à chaque couche lors de la création de celle-ci.

Pour plus d'informations sur les groupes de sécurité, consultez Utilisation des groupes de sécurité.

Une fois que la couche a été créée, vous pouvez utiliser Security Groups pour ajouter des groupes de sécurité à la couche en les sélectionnant dans la liste Custom security groups. Après avoir ajouté un groupe de sécurité à une couche, AWS OpsWorks Stacks l'ajoute à toutes les nouvelles instances. (Notez que les instances du magasin d'instances redémarrées seront affichées en tant que nouvelles instances, de sorte qu'elles comporteront également les nouveaux groupes de sécurité.) AWS OpsWorks Stacks n'ajoute pas de groupes de sécurité aux instances en ligne.

Vous pouvez supprimer les groupes de sécurité existants en cliquant sur le x, comme suit :

- Si vous avez choisi que AWS OpsWorks Stacks associe automatiquement les groupes de sécurité intégrés, vous pouvez supprimer les groupes de sécurité personnalisés que vous avez ajoutés précédemment en cliquant sur le x, mais vous ne pouvez pas supprimer le groupe intégré.
- Si vous avez choisi de ne pas associer automatiquement les groupes de sécurité, vous pouvez supprimer les groupes de sécurité existants, y compris celui d'origine, tant que la couche conserve au moins un groupe.

Une fois que vous avez supprimé un groupe de sécurité d'une couche, AWS OpsWorks Stacks ne l'ajoute à aucune instance nouvelle ou redémarrée. AWS OpsWorks Stacks ne supprime pas les groupes de sécurité des instances en ligne.

> **Note**
>
> Si votre stack s'exécute dans un VPC, vous pouvez ajouter ou supprimer un groupe de sécurité pour une instance en ligne à l'aide de la EC2 console, de l'API ou de la CLI

Amazon. Toutefois, ce groupe de sécurité ne sera pas visible dans la console AWS OpsWorks Stacks. Si vous souhaitez supprimer le groupe de sécurité, vous devez également utiliser Amazon EC2. Pour plus d'informations, consultez Groupes de sécurité.

Remarques :

- Vous ne pouvez pas limiter les paramètres d'accès au port d'un groupe de sécurité intégré en ajoutant un groupe de sécurité plus restrictif. Lorsqu'il existe plusieurs groupes de sécurité, Amazon EC2 utilise les paramètres les plus permissifs.
- Vous ne devez pas modifier la configuration d'un groupe de sécurité intégré. Lorsque vous créez une pile, AWS OpsWorks Stacks remplace les configurations des groupes de sécurité intégrés, de sorte que toutes les modifications que vous apportez seront perdues lors de la prochaine création d'une pile.

Si vous découvrez que vous avez besoin de paramètres de groupe de sécurité plus restrictifs pour une ou plusieurs couches, procédez comme suit :

1. Créez des groupes de sécurité personnalisés avec les paramètres appropriés et ajoutez-les aux couches appropriées.

   Chaque couche de votre pile doit avoir au moins un groupe de sécurité en plus du groupe intégré, même si une seule couche nécessite des paramètres personnalisés.
2. Modifiez la configuration de la pile et réglez le paramètre Utiliser les groupes de OpsWorks sécurité sur Non.

   AWS OpsWorks Stacks supprime automatiquement le groupe de sécurité intégré de chaque couche.

Pour plus d'informations sur les groupes de sécurité, consultez Amazon EC2 Security Groups.

EC2 Profil d'instance

Vous pouvez modifier le EC2 profil des instances de la couche. Pour de plus amples informations, veuillez consulter Spécification des autorisations pour les applications exécutées sur EC2 des instances.

CloudWatch Journaux

L'onglet CloudWatch Logs vous permet d'activer ou de désactiver Amazon CloudWatch Logs. CloudWatch L'intégration des journaux fonctionne avec les stacks basés sur Linux Chef 11.10

et Chef 12. Pour plus d'informations sur l'activation CloudWatch de l'intégration des journaux et sur la spécification des journaux que vous souhaitez gérer dans la console CloudWatch Logs, consultezUtilisation d'Amazon CloudWatch Logs avec AWS OpsWorks Stacks.

Balises

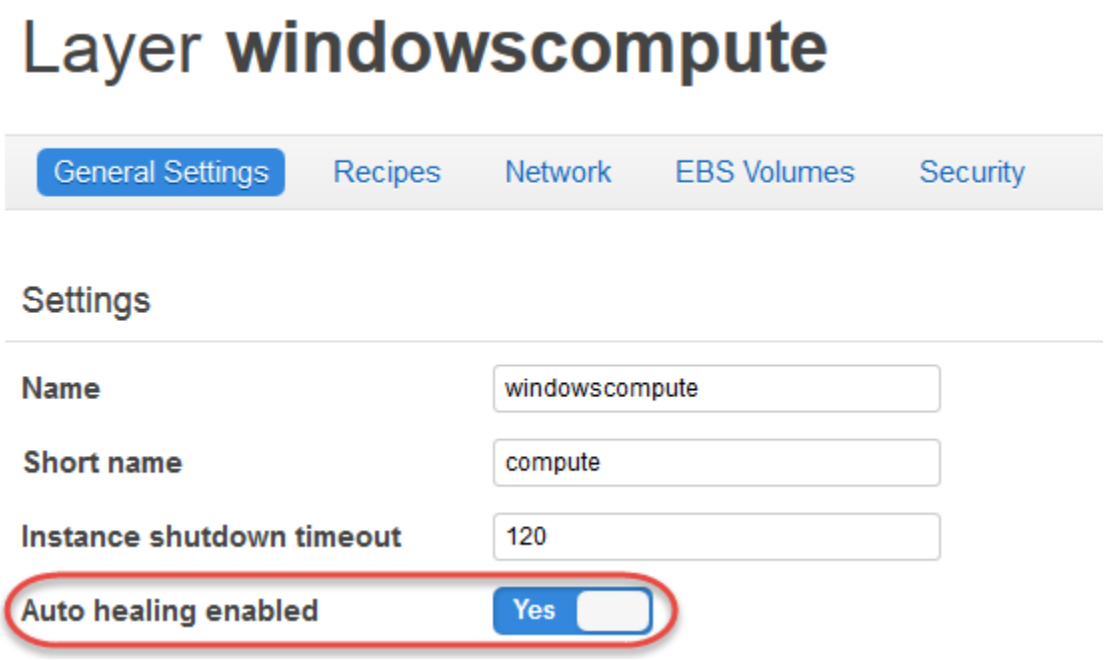
L'onglet Balises vous permet d'appliquer des balises de répartition des coûts à votre couche. Après avoir ajouté des tags, vous pouvez les activer dans la AWS Billing and Cost Management console. Lorsque vous créez une balise, vous appliquez la balise à chaque ressource au sein de la structure balisée. Par exemple, si vous appliquez une balise à une couche, vous l'appliquez à chaque instance, volume Amazon EBS ou équilibreur de charge Elastic Load Balancing de la couche. Pour plus d'informations sur la façon d'activer vos tags et de les utiliser pour suivre et gérer les coûts de vos ressources AWS OpsWorks Stacks, consultez les sections Utilisation des balises de répartition des coûts et activation des balises de répartition des coûts définies par l'utilisateur dans le guide de l'utilisateur de Billing and Cost Management. Pour plus d'informations sur le balisage dans AWS OpsWorks Stacks, consultez Balises.

Utilisation de la réparation automatique pour remplacer les instances en échec

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Chaque instance dispose d'un agent AWS OpsWorks Stacks qui communique régulièrement avec le service. AWS OpsWorks Stacks utilise cette communication pour surveiller l'état de santé de l'instance. Si un agent ne communique pas avec le service pendant plus de cinq minutes environ, AWS OpsWorks Stacks considère que l'instance a échoué.

La réparation automatique est définie au niveau de la couche ; vous pouvez changer le paramètre de réparation automatique en modifiant les paramètres de la couche, comme illustré dans la capture d'écran suivante.



> **Note**
>
> Une instance peut être membre de plusieurs couches. Si la guérison automatique est désactivée sur l'une de ces couches, AWS OpsWorks Stacks ne répare pas l'instance en cas d'échec.

Si la réparation automatique est activée sur une couche (paramètre par défaut),AWS OpsWorks Stacks remplace automatiquement les instances défaillantes de la couche comme suit :

Instance basée sur le stockage d'instance

1. Arrête l' EC2 instance Amazon et vérifie qu'elle s'est arrêtée.
2. Supprime les données sur le volume racine.
3. Crée une nouvelle EC2 instance Amazon avec le même nom d'hôte, la même configuration et la même appartenance à la couche.
4. Réattache tous les volumes Amazon EBS, y compris les volumes attachés après le démarrage initial de l'ancienne instance.
5. Attribue une nouvelle adresse IP publique et privée.
6. Si l'ancienne instance était associée à une adresse IP Elastic, associe la nouvelle instance à la même adresse IP.

Instance basée sur les volumes Amazon EBS

1. Arrête l' EC2 instance Amazon et vérifie qu'elle s'est arrêtée.
2. Démarre l' EC2 instance.

Une fois que l'instance réparée automatiquement est de nouveau en ligne, AWS OpsWorks Stacks déclenche un événement de configuration du cycle de vie sur toutes les instances de la pile. Les attributs de configuration et de déploiement de la pile associés incluent les adresses IP publiques et privées de l'instance. Les recettes Configure personnalisées peuvent obtenir les nouvelles adresses IP à partir de l'objet de nœud.

Si vous spécifiez un volume Amazon EBS pour les instances d'une couche, AWS OpsWorks Stacks crée un nouveau volume et l'attache à chaque instance au démarrage de l'instance. Si vous souhaitez ensuite détacher le volume à partir d'une instance, utilisez la page Ressources.

Lorsque AWS OpsWorks Stacks soigne automatiquement l'une des instances d'une couche, il gère les volumes de la manière suivante :

- Si le volume était attaché à l'instance lorsque celle-ci a échoué, le volume et ses données sont enregistrés, et AWS OpsWorks Stacks l'attache à la nouvelle instance.
- Si le volume n'était pas attaché à l'instance lorsqu'elle a échoué, AWS OpsWorks Stacks crée un volume vide avec la configuration spécifiée par la couche et attache ce volume à la nouvelle instance.

La réparation automatique est activée par défaut pour toutes les couches, mais vous pouvez modifier les paramètres généraux de la couche pour la désactiver.

> **Important**
>
> Si la réparation automatique est activée, veillez à effectuer les opérations suivantes :
>
> - Utilisez uniquement la console, la CLI ou l'API AWS OpsWorks Stacks pour arrêter les instances.
>
> Si vous arrêtez une instance d'une autre manière, par exemple en utilisant la EC2 console Amazon, AWS OpsWorks Stacks considère l'instance comme défaillante et la répare automatiquement.

- Utilisez les volumes Amazon EBS pour stocker les données que vous ne voulez pas perdre si l'instance est réparée automatiquement.

  La réparation automatique arrête l'ancienne EC2 instance Amazon, qui détruit toutes les données qui ne sont pas stockées sur un volume Amazon EBS. Les volumes Amazon EBS sont rattachés à la nouvelle instance, qui préserve toutes les données stockées.

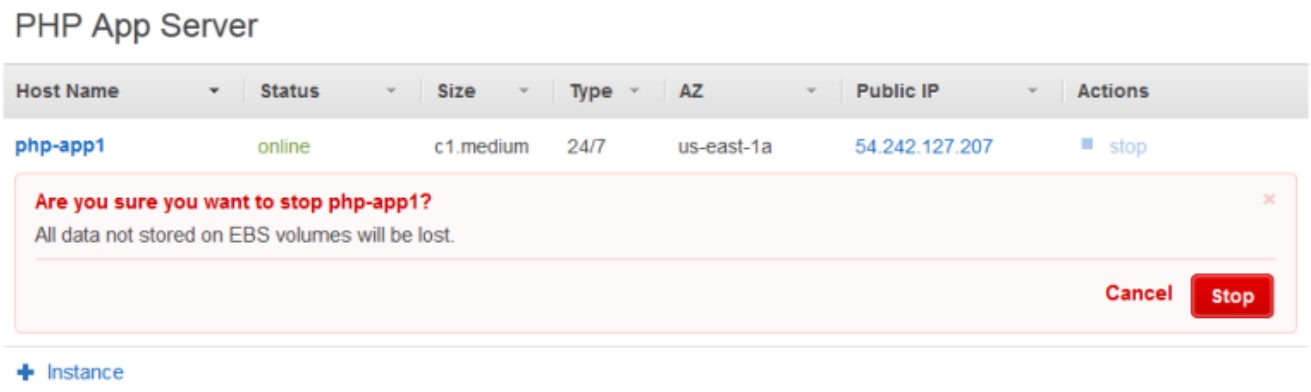
Supprimer une OpsWorks couche

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

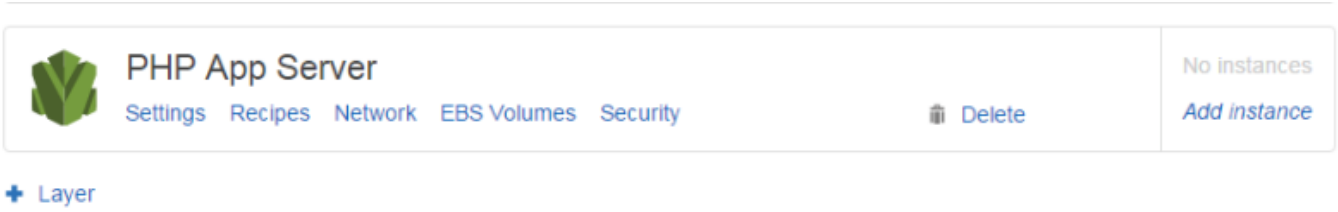
Si vous n'avez plus besoin d'une couche AWS OpsWorks Stacks, vous pouvez la supprimer de votre pile.

Pour supprimer une OpsWorks couche

1. Dans le volet de navigation, cliquez sur Instances.
2. Sur la page Instances, sous le nom de la couche que vous souhaitez supprimer, cliquez sur stop (arrêter) dans la colonne Actions pour chaque instance.



3. Une fois que chaque instance est arrêtée, cliquez sur delete (supprimer) pour la supprimer de la couche.
4. Dans le volet de navigation, cliquez sur Layers.
5. Sur la page Couches, choisissez Supprimer.



Couche Elastic Load Balancing

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Le fonctionnement d'Elastic Load Balancing est légèrement différent de celui d'une couche AWS OpsWorks Stacks. Au lieu de créer une couche et d'y ajouter des instances, vous utilisez la console ou l'API Elastic Load Balancing pour créer un équilibreur de charge, puis l'associer à une couche existante. Outre la distribution du trafic vers les instances de la couche, Elastic Load Balancing effectue les opérations suivantes :

- Détecte les EC2 instances Amazon non saines et redirige le trafic vers les instances saines restantes jusqu'à ce que les instances non saines soient restaurées.
- Dimensionne automatiquement la capacité de traitement des demandes en réponse au trafic entrant.
- Si vous activez le drainage de la connexion, l'équilibreur de charge cesse d'envoyer de nouvelles demandes aux instances non saines ou dont l'enregistrement est sur le point d'être annulé, mais maintient la connexion, jusqu'au délai d'attente spécifié, pour permettre à l'instance de terminer toutes les requêtes en cours.

Après avoir attaché un équilibreur de charge à une couche, AWS OpsWorks Stacks effectue les opérations suivantes :

- Annule l'enregistrement de toutes les instances actuellement enregistrées.
- Enregistre automatiquement les instances de la couche lorsqu'elles sont en ligne et annule l'enregistrement des instances quand elles passent hors connexion, y compris les instances à date définie et les instances à charge définie.
- Commence automatiquement à acheminer les requêtes vers les instances enregistrées dans leurs zones de disponibilité.

Si vous avez activé la fonction de vidange des connexions de l'équilibreur de charge, vous pouvez spécifier si AWS OpsWorks Stacks la prend en charge. Si vous activez la prise en charge du drainage des connexions (paramètre par défaut), après l'arrêt d'une instance, AWS OpsWorks Stacks effectue les opérations suivantes :

- Annule l'enregistrement de l'instance auprès de l'équilibreur de charge.

  L'équilibreur de charge cesse d'envoyer de nouvelles demandes et démarre le drainage de la connexion.
- Retarde le déclenchement d'un événement de cycle de vie Shutdown jusqu'à ce que l'équilibreur de charge ait terminé le drainage de la connexion.

Si vous n'activez pas la prise en charge de l'épuisement des connexions, AWS OpsWorks Stacks déclenche l'événement Shutdown dès que l'instance est arrêtée, même si l'instance est toujours connectée à l'équilibreur de charge.

Pour utiliser Elastic Load Balancing avec une pile, vous devez d'abord créer un ou plusieurs équilibreurs de charge dans la même région à l'aide de la console, de la CLI ou de l'API Elastic Load Balancing. Vous devez être conscient des points suivants :

- Vous ne pouvez attacher qu'un seul équilibreur de charge à une couche.
- Chaque équilibreur de charge ne peut gérer qu'une seule couche.
- AWS OpsWorks Stacks ne prend pas en charge Application Load Balancer. Vous ne pouvez utiliser Classic Load Balancer qu'avec AWS OpsWorks Stacks.

Cela signifie que vous devez créer un équilibreur de charge Elastic Load Balancing distinct pour chaque couche de chaque pile que vous souhaitez équilibrer et l'utiliser uniquement à cette fin. Il est recommandé d'attribuer un nom distinct à chaque équilibreur de charge Elastic Load Balancing que vous prévoyez d'utiliser avec AWS OpsWorks Stacks, tel que MyStack 1- RailsLayer -ELB, afin d'éviter d'utiliser un équilibreur de charge à plusieurs fins.

> **Important**
>
> Nous vous recommandons de créer de nouveaux équilibreurs de charge Elastic Load Balancing pour vos couches AWS OpsWorks Stacks. Si vous choisissez d'utiliser un équilibreur de charge Elastic Load Balancing existant, vous devez d'abord vérifier qu'il n'est pas utilisé à d'autres fins et qu'aucune instance n'est attachée. Une fois que l'équilibreur de charge est attaché à la couche, il OpsWorks supprime toutes les instances existantes et configure l'équilibreur de charge pour qu'il gère uniquement les instances de la couche. Bien qu'il soit techniquement possible d'utiliser la console ou l'API Elastic Load Balancing pour modifier la configuration d'un équilibreur de charge après l'avoir attaché à une couche, vous ne devez pas le faire ; les modifications ne seront pas permanentes.

Pour attacher un équilibreur de charge Elastic Load Balancing à une couche

1. Si ce n'est pas encore fait, utilisez la console, l'API ou la CLI Elastic Load Balancing pour créer un équilibreur de charge dans la région de la pile. Lorsque vous créez l'équilibreur de charge, procédez comme suit :

   - Assurez-vous de spécifier un chemin ping de vérification du statut approprié à votre application.

     Comme le chemin d'accès ping par défaut est `/index.html`, si la racine de votre application n'inclut pas `index.html`, vous devez spécifier un chemin d'accès ping approprié, sans quoi la vérification du statut échoue.
   - Si vous souhaitez utiliser le drainage de la connexion, assurez-vous que la fonctionnalité est activée et qu'elle possède une valeur de délai appropriée.

   Pour plus d'informations, consultez Elastic Load Balancing.
2. Créez la couche à équilibrer ou modifiez les paramètres réseau d'une couche existante.

> **Note**
>
> Vous ne pouvez pas attacher un équilibreur de charge lorsque vous créez une couche personnalisée. Vous devez modifier les paramètres de la couche.

3. Sous Elastic Load Balancing, sélectionnez l'équilibreur de charge que vous souhaitez attacher à la couche et spécifiez si vous souhaitez que AWS OpsWorks Stacks prenne en charge le drainage des connexions.

Une fois que vous avez attaché un équilibreur de charge à une couche, AWS OpsWorks Stacks déclenche un événement de configuration du cycle de vie sur les instances de la pile pour les informer de la modification. AWS OpsWorks Stacks déclenche également un événement Configure lorsque vous détachez un équilibreur de charge.

> **Note**
>
> Après le démarrage d'une instance, AWS OpsWorks Stacks exécute les recettes de configuration et de déploiement, qui installent des packages et déploient des applications. Une fois ces recettes terminées, l'instance est en ligne et AWS OpsWorks Stacks l'enregistre auprès d'Elastic Load Balancing. AWS OpsWorks Stacks déclenche également un événement Configure après la mise en ligne de l'instance. Cela signifie que l'enregistrement d'Elastic Load Balancing et les recettes de configuration peuvent s'exécuter simultanément, et que l'instance peut être enregistrée avant la fin des recettes de configuration. Pour garantir qu'une recette se termine avant qu'une instance ne soit enregistrée auprès d'Elastic Load Balancing, vous devez ajouter la recette aux événements du cycle de vie de configuration ou de déploiement de la couche. Pour de plus amples informations, veuillez consulter Exécution des recettes.

Il est parfois utile de supprimer une instance d'un équilibreur de charge. Par exemple, lorsque vous mettez à jour une application, nous vous recommandons de déployer l'application sur une seule instance et de vérifier que l'application fonctionne correctement avant de la déployer sur chaque instance. Généralement, comme vous supprimez cette instance à partir de l'équilibreur de charge, il ne reçoit pas les demandes de l'utilisateur jusqu'à ce que vous ayez vérifié la mise à jour.

Vous devez utiliser la console ou l'API Elastic Load Balancing pour supprimer temporairement une instance en ligne d'un équilibreur de charge. Ce qui suit explique comment utiliser la console.

Pour supprimer temporairement une instance d'un équilibreur de charge

1. Ouvrez la EC2 console Amazon et choisissez Load Balancers.
2. Choisissez l'équilibreur de charge approprié et ouvrez l'onglet Instances.
3. Choisissez Remove from Load Balancer dans la colonne Actions de l'instance.
4. Lorsque vous avez terminé, choisissez Edit Instances et renvoyez l'instance à l'équilibreur de charge.

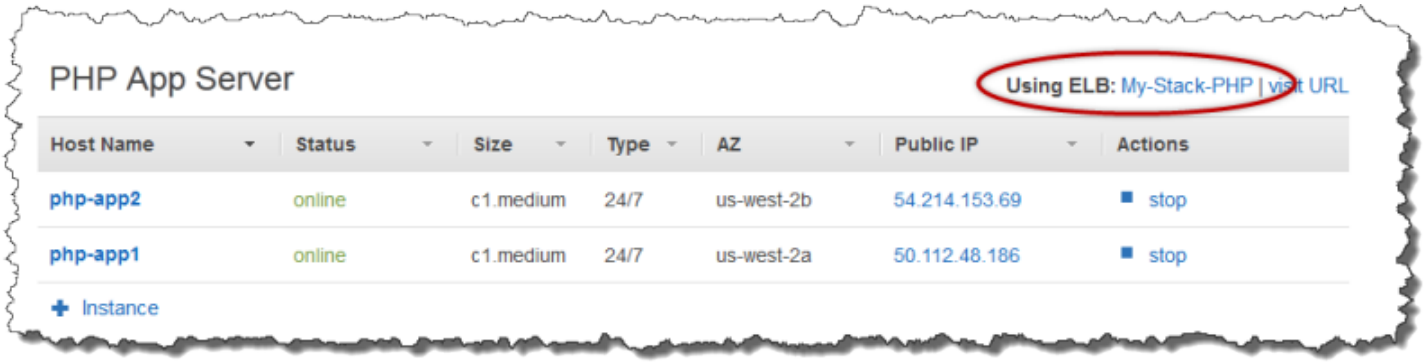
> **Important**
>
> Si vous utilisez la console ou l'API Elastic Load Balancing pour supprimer une instance d'un équilibreur de charge, vous devez également utiliser Elastic Load Balancing pour la remettre en place. AWS OpsWorks Stacks n'est pas au courant des opérations que vous effectuez avec d'autres consoles de service et ne renverra pas l'instance dans l'équilibreur de charge à votre place. APIs Parfois, AWS OpsWorks Stacks peut réajouter l'instance à l'ELB, mais ce comportement n'est pas garanti et ne se produit pas dans tous les cas.

Vous pouvez attacher plusieurs équilibreurs de charge à un ensemble particulier d'instances comme suit :

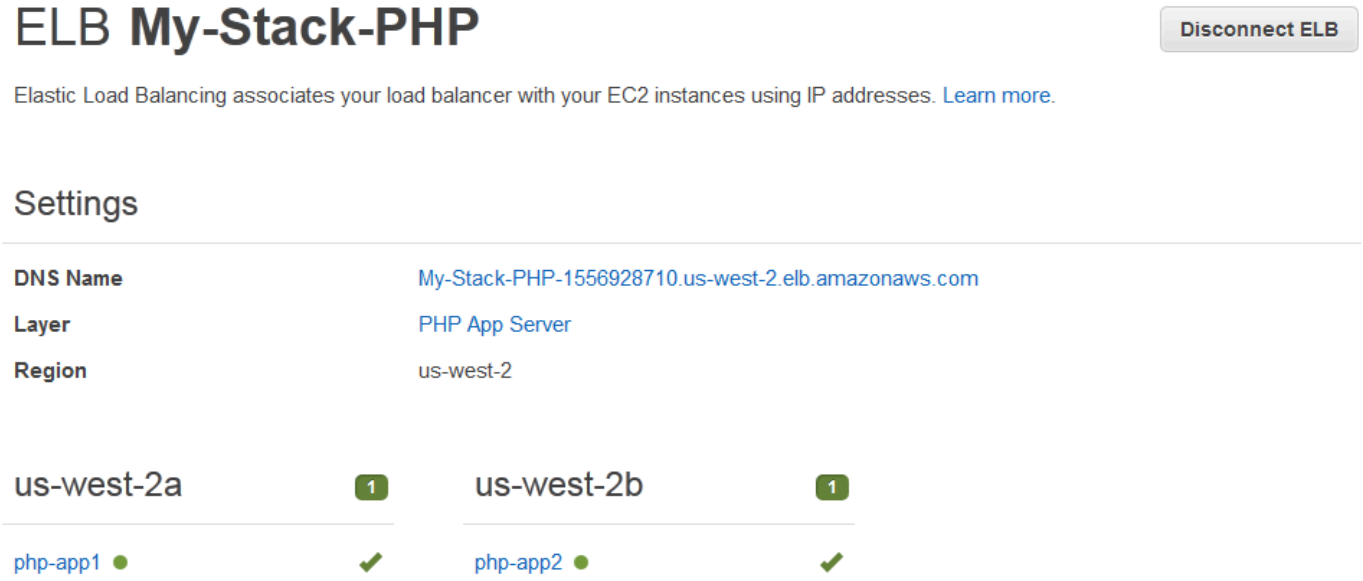
Pour attacher plusieurs équilibreurs de charge

1. Utilisez la console, l'API ou la CLI Elastic Load Balancing pour créer un ensemble d'équilibreurs de charge.
2. Créez une couche personnalisée pour chaque équilibreur de charge et attachez-lui l'un des équilibreurs de charge. Vous n'avez pas besoin d'implémenter de recettes personnalisées pour ces couches ; une couche personnalisée par défaut suffit.
3. Ajoutez l'ensemble des instances à chaque couche personnalisée.

Vous pouvez examiner les propriétés d'un équilibreur de charge en accédant à la page Instances et en cliquant sur le nom de l'équilibreur de charge approprié.



La page ELB affiche les propriétés de base de l'équilibreur de charge, y compris son nom DNS et l'état de santé des instances associées. Si la pile s'exécute dans un VPC, la page affiche les sous-réseaux plutôt que les zones de disponibilité. Une coche verte indique une instance saine. Vous pouvez cliquer sur le nom pour vous connecter à un serveur, via l'équilibreur de charge.



Couche de service Amazon RDS

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une couche de service Amazon RDS représente une instance Amazon RDS. La couche ne peut représenter que les instances Amazon RDS existantes, que vous devez créer séparément à l'aide de la console ou de l'API Amazon RDS.

La procédure de base pour intégrer une couche de service Amazon RDS dans votre stack est la suivante :

1. Utilisez la console, l'API ou la CLI Amazon RDS pour créer une instance.

   Veillez à enregistrer l'ID, le nom d'utilisateur principal, le mot de passe principal et le nom de base de données de l'instance.
2. Pour ajouter une couche Amazon RDS à votre pile, enregistrez l'instance Amazon RDS auprès de la pile.
3. Attachez la couche à une application, ce qui ajoute les informations de connexion de l'instance Amazon RDS aux `deploy`attributs de l'application.
4. Utilisez les fichiers spécifiques à la langue ou les informations contenues dans les `deploy` attributs pour connecter l'application à l'instance Amazon RDS.

   Pour plus d'informations sur la connexion d'une application à un serveur de base de données, consultez the section called “Connexion à une base de données”

> **Warning**
>
> Veillez à ce que les caractères dans le nom d'utilisateur et le mot de passe principaux de l'instance soient compatibles avec votre serveur d'applications. Par exemple, dans le cas de la couche Java App Server, l'inclusion & dans l'une ou l'autre des chaînes provoque une erreur d'analyse XML qui empêche le serveur Tomcat de démarrer.

Rubriques

- Spécification des groupes de sécurité
- Enregistrement d'une instance Amazon RDS auprès d'une pile
- Associer des couches de service Amazon RDS à des applications
- Supprimer une couche de service Amazon RDS d'une pile

Spécification des groupes de sécurité

Pour utiliser une instance Amazon RDS avec AWS OpsWorks Stacks, la base de données ou les groupes de sécurité VPC doivent autoriser l'accès depuis les adresses IP appropriées. En production, un groupe de sécurité limite généralement l'accès uniquement aux adresses IP qui ont besoin d'accéder à la base de données. Il inclut généralement les adresses des systèmes que vous utilisez pour gérer la base de données et les instances AWS OpsWorks Stacks qui doivent accéder à la base de données. AWS OpsWorks Stacks crée automatiquement un groupe EC2 de sécurité Amazon pour chaque type de couche lorsque vous créez votre première pile dans une région. Un moyen simple de fournir un accès aux instances AWS OpsWorks Stacks consiste à attribuer les groupes de sécurité AWS OpsWorks Stacks appropriés à l'instance Amazon RDS ou au VPC.

Pour spécifier des groupes de sécurité pour une instance Amazon RDS existante

1. Ouvrez la console Amazon RDS à l'adresse https://console.aws.amazon.com/rds/.
2. Cliquez sur Instances dans le volet de navigation et sélectionnez l'instance Amazon RDS appropriée. Cliquez sur Actions d'instance, Modifier.
3. Sélectionnez les groupes de sécurité suivants dans la liste Groupe de sécurité, puis cliquez sur Continuer et Modifier l'instance DB afin de mettre à jour l'instance.

   - Le groupe de sécurité AWS- OpsWorks -DB-master-server (). *`security_group_id`*
   - Le groupe de sécurité pour la couche de serveur d'applications dont les instances doivent se connecter à la base de données. Le nom du groupe inclut le nom de la couche. Par exemple, pour fournir un accès à la base de données aux instances de PHP App Server, spécifiez le groupe AWS- OpsWorks -PHP-App-Server.

Si vous créez une nouvelle instance Amazon RDS, vous pouvez spécifier les groupes de sécurité AWS OpsWorks Stacks appropriés sur la page Configurer les paramètres avancés de l'assistant de lancement d'une instance de base de données. Pour une description de l'utilisation de cet assistant, consultez Création d'une instance de base de données MySQL et connexion à une base de données sur une instance de base de données MySQL.

Pour plus d'informations sur la spécification des groupes de sécurité VPC, consultez Groupes de sécurité pour le VPC.

Enregistrement d'une instance Amazon RDS auprès d'une pile

Pour ajouter une couche de service Amazon RDS dans une pile, vous devez enregistrer une instance auprès de la pile.

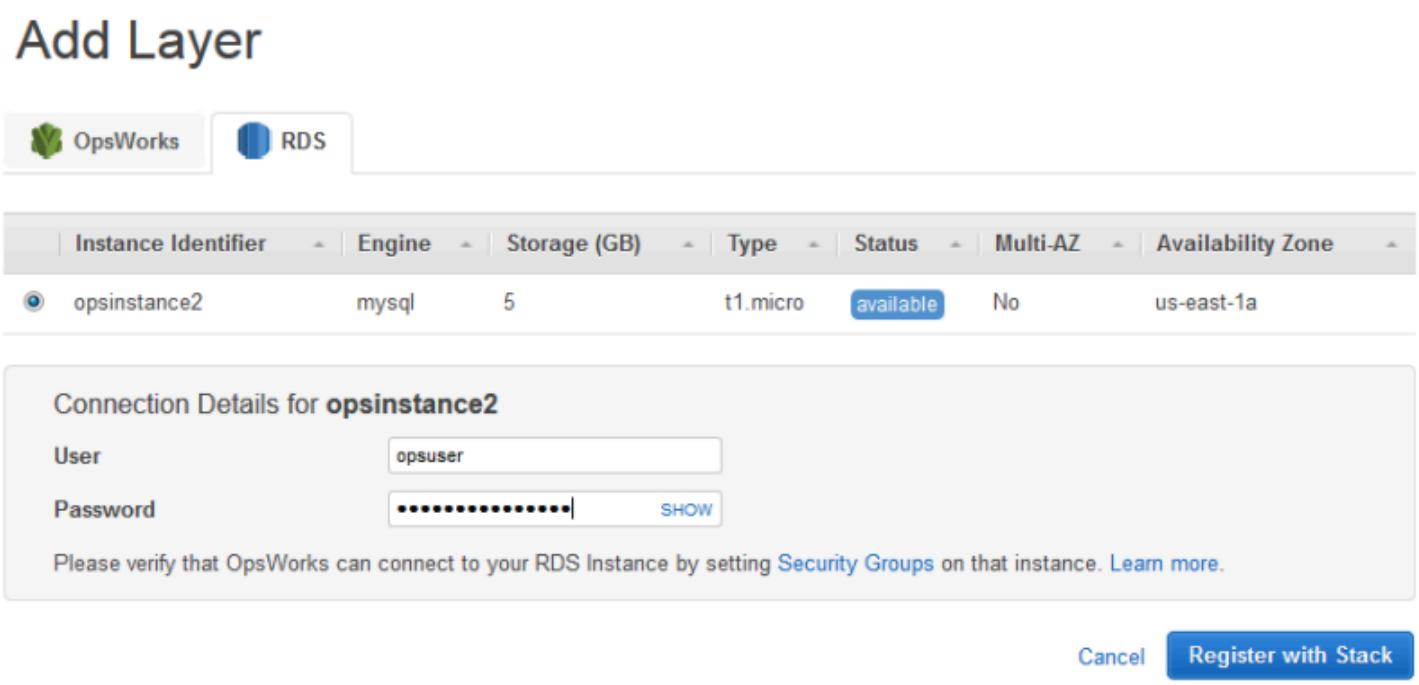
Pour enregistrer une instance Amazon RDS auprès d'une pile

1. Dans la console AWS OpsWorks Stacks, cliquez sur Couche dans le volet de navigation, sur + Layer ou sur Ajouter une couche pour ouvrir la page Ajouter une couche, puis sur l'onglet RDS.
2. Si nécessaire, mettez à jour le rôle de service de la pile, comme décrit dans Mise à jour de rôle de service de la pile.
3. Cliquez sur l'onglet RDS pour répertorier les instances Amazon RDS disponibles.

> **Note**
>
> Si votre compte ne possède aucune instance Amazon RDS, vous pouvez en créer une en cliquant sur Ajouter une instance RDS dans l'onglet RDS, ce qui vous amène à la console Amazon RDS et lance l'assistant de lancement d'une instance de base de données. Vous pouvez également accéder directement à la console Amazon RDS et cliquer sur Lancer une instance de base de données, ou utiliser l'API ou la CLI Amazon RDS. Pour plus d'informations sur la façon de créer une instance Amazon RDS, consultez Mise en route sur Amazon RDS.

4. Sélectionnez l'instance appropriée, définissez Utilisateur et Mot de passe sur les valeurs appropriées d'utilisateur et de mot de passe, puis cliquez sur Register to Stack (Enregistrer sur Stack).

> **Important**
>
> Vous devez vous assurer que l'utilisateur et le mot de passe que vous utilisez pour enregistrer l'instance Amazon RDS correspondent à un utilisateur et à un mot de passe valides. Dans le cas contraire, vos applications ne pourront pas se connecter à l'instance. Cependant, vous pouvez modifier la couche afin de fournir des valeurs valides d'utilisateur et de mot de passe, puis redéployer l'application.



Lorsque vous ajoutez une couche de service Amazon RDS à une pile, AWS OpsWorks Stacks lui attribue un ID et ajoute la configuration Amazon RDS associée à la configuration de la pile et à l'attribut de déploiement. `[:opsworks][:stack]`

> **Note**
>
> Si vous modifiez le mot de passe d'une instance Amazon RDS enregistrée, vous devez le mettre à jour manuellement dans AWS OpsWorks Stacks, puis redéployer vos applications pour mettre à jour la configuration de la pile et les attributs de déploiement sur les instances de la pile.

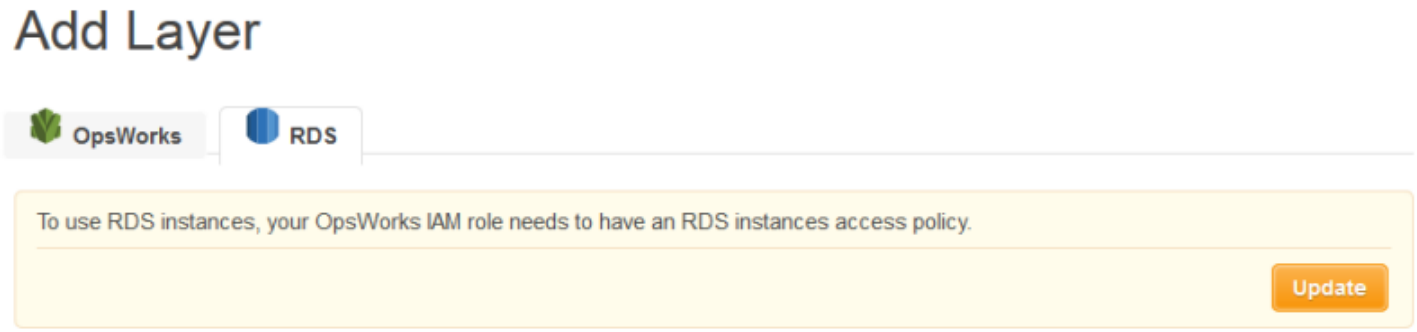
Rubriques

- Mise à jour de rôle de service de la pile

Mise à jour de rôle de service de la pile

Chaque stack possède un rôle de service IAM qui spécifie les actions que AWS OpsWorks Stacks peut effectuer en votre nom avec les autres services AWS. Pour enregistrer une instance Amazon RDS auprès d'une pile, son rôle de service doit accorder à AWS OpsWorks Stacks l'autorisation d'accéder à Amazon RDS.

La première fois que vous ajoutez une couche de service Amazon RDS à l'une de vos piles, le rôle de service peut ne pas disposer des autorisations requises. Si oui, lorsque vous cliquez sur l'onglet RDS de la page Add Layer (Ajouter une couche), vous voyez ce qui suit.



Cliquez sur Mettre à jour pour que AWS OpsWorks Stacks mette à jour la politique du rôle de service comme suit.

```
{"Statement": [{"Action": ["ec2:*", "iam:PassRole",
                       "cloudwatch:GetMetricStatistics",
                       "elasticloadbalancing:*",
                       "rds:*"],
            "Effect": "Allow",
            "Resource": ["*"] }]
}
```

> **Note**
>
> Vous devez effectuer la mise à jour une seule fois. Le rôle mis à jour est ensuite utilisé automatiquement par l'ensemble de vos piles.

Associer des couches de service Amazon RDS à des applications

Après avoir ajouté une couche de service Amazon RDS, vous pouvez l'associer à une application.

- Vous pouvez associer une couche Amazon RDS à une application lorsque vous créez l'application, ou ultérieurement en modifiant la configuration de l'application.
- Pour dissocier une couche Amazon RDS d'une application, modifiez la configuration de l'application pour spécifier un autre serveur de base de données, ou aucun serveur.

La couche Amazon RDS fait toujours partie de la pile et peut être associée à une autre application.

Après avoir associé une instance Amazon RDS à une application, AWS OpsWorks Stacks place les informations de connexion à la base de données sur les serveurs de l'application. L'application sur chaque instance de serveur peut ensuite utiliser ces informations afin de se connecter à la base de données. Pour plus d'informations sur la connexion à une instance Amazon RDS, consultezthe section called “Connexion à une base de données”.

Supprimer une couche de service Amazon RDS d'une pile

Pour supprimer une couche de service Amazon RDS d'une pile, vous devez la désenregistrer.

Pour annuler l'enregistrement d'une couche de service Amazon RDS

1. Cliquez sur Layers dans le volet de navigation, puis sur le nom de la couche de service Amazon RDS.
2. Cliquez sur Annuler l'inscription et confirmez que vous souhaitez annuler l'enregistrement de la couche.

Cette procédure supprime la couche de la pile, mais elle ne supprime pas l'instance Amazon RDS sous-jacente. L'instance et les bases de données restent dans votre compte et peuvent être enregistrés auprès d'autres piles. Vous devez utiliser la console Amazon RDS, une API ou une CLI pour supprimer l'instance. Pour plus d'informations, consultez Suppression d'une instance de base de données.

Couches de cluster ECS

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Le service Amazon Elastic Container Service (Amazon ECS) gère les conteneurs Docker dans un cluster d'instances Amazon Elastic Compute Cloud (Amazon EC2), appelées instances de conteneurs. Une couche de cluster ECS représente un cluster Amazon ECS et simplifie la gestion du cluster en fournissant des fonctionnalités telles que :

- Mise en service et gestion rationalisées de l'instance de conteneur
- Mises à jour des packages et du système d'exploitation des instances de conteneur
- Gestion des autorisations utilisateur
- Surveillance des performances des instances de conteneur
- Gestion des volumes Amazon Elastic Block Store (Amazon EBS)
- Gestion des adresses IP publiques et Elastic
- Gestion des groupes de sécurité

La couche ECS Cluster comporte les restrictions et exigences suivantes :

- La couche est disponible uniquement pour les piles Chef 11.10 Chef 12 Linux exécutées dans un VPC, y compris un VPC par défaut.
- Les instances de la couche doivent exécuter l'un des systèmes d'exploitation suivants :
  - 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
  - Personnalisé
- La version de l'agent AWS OpsWorks Stacks de l'instance de couche doit être `3425-20150727112318` ou une version ultérieure.

Rubriques

- Ajouter une couche de cluster ECS à une pile
- Gestion du cluster ECS
- Supprimer une couche de cluster ECS d'une pile

Ajouter une couche de cluster ECS à une pile

AWS OpsWorks Stacks simplifie le processus de lancement et de maintenance des instances de conteneur pour les clusters Amazon ECS existants. Pour créer ou lancer d'autres entités Amazon ECS, comme des clusters et des tâches, utilisez la console, l'interface de ligne de commande (CLI) Amazon ECS. (Pour plus d'informations, consultez le Manuel du développeur Amazon Elastic Container Service.) Vous pouvez ensuite associer un cluster à une pile en créant une couche de cluster ECS, que vous pouvez utiliser pour gérer le cluster dans AWS OpsWorks Stacks.

Vous pouvez associer des clusters aux piles comme suit :

- Chaque pile peut comporter une couche de cluster ECS, qui représente un seul cluster.
- Un cluster peut être associé à un seule pile.

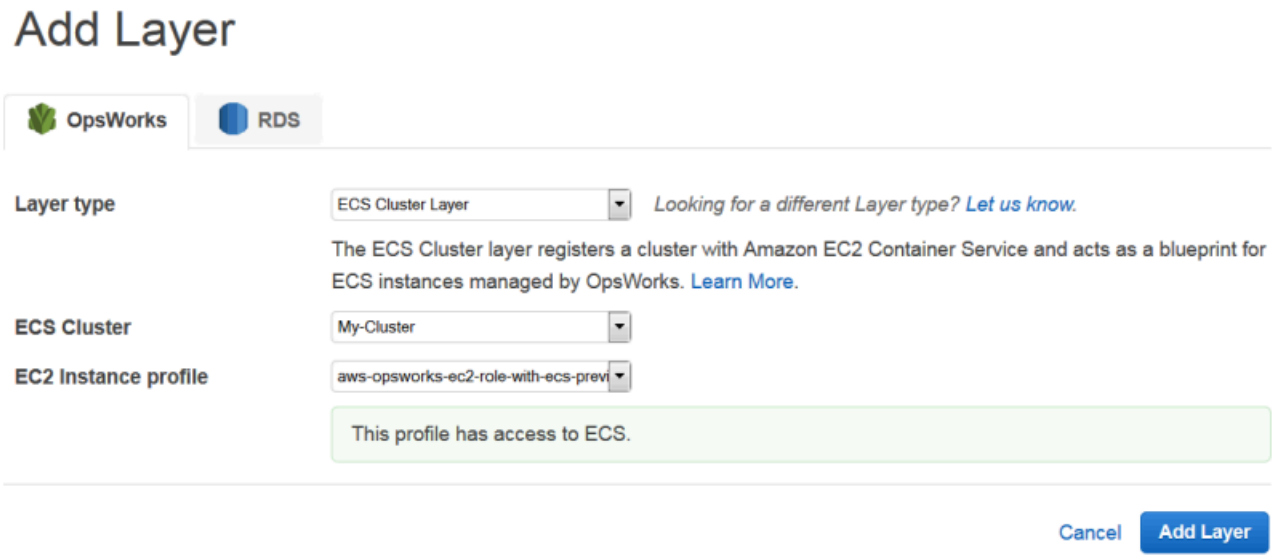
Avant de pouvoir ajouter des couches de cluster ECS à vos piles, vous devez mettre à jour le rôle de service AWS OpsWorks Stacks AWS Identity and Access Management (IAM), généralement nommé`aws-opsworks-service-role`, afin de permettre à AWS OpsWorks Stacks d'interagir avec Amazon ECS en votre nom. Pour plus d'informations sur le rôle de service, consultez Permettre à AWS OpsWorks Stacks d'agir en votre nom.

La première fois que vous créez une couche de cluster ECS, la console fournit un bouton de mise à jour que vous pouvez choisir pour demander à AWS OpsWorks Stacks de mettre à jour le rôle pour vous. AWS OpsWorks Stacks affiche ensuite la page Ajouter une couche afin que vous puissiez ajouter la couche à la pile. Vous devez mettre à jour le rôle de service une seule fois. Vous pouvez ensuite utiliser le rôle mis à jour pour ajouter une couche de cluster ECS à n'importe quelle pile.

> **Note**
>
> Si vous le préférez, vous pouvez mettre à jour manuellement la stratégie du rôle de service en ajoutant l'autorisation `ecs:*` à la stratégie existante, comme suit :

```
{
  "Statement": [
    {
      "Action": [
        "ec2:*",
        "iam:PassRole",
        "cloudwatch:GetMetricStatistics",
        "elasticloadbalancing:*",
        "rds:*",
        "ecs:*"
      ],
      "Effect": "Allow",
      "Resource": ["*"]
    }
  ]
}
```

L'association d'un cluster à une pile requiert deux opérations : l'inscription du cluster avec la pile, puis la création de la couche associée. La console AWS OpsWorks Stacks combine ces étapes ; la création de couches enregistre automatiquement le cluster spécifié. Si vous utilisez l'API, la CLI ou le SDK AWS OpsWorks Stacks, vous devez effectuer des opérations distinctes pour enregistrer le cluster et créer la couche associée. Pour utiliser la console afin d'ajouter une couche de cluster ECS à votre pile, choisissez Layers, choisissez +Layer ou Add a Layer, puis choisissez le type de couche de cluster ECS.



La page Add Layer (Ajouter une couche) inclut les options de configuration suivantes :

Cluster ECS

Le cluster Amazon ECS que vous souhaitez enregistrer auprès de la pile.

EC2 Profil de l'instance

Profil d'instance Amazon Elastic Compute Cloud (Amazon EC2) du cluster. Ce profil autorise les applications exécutées sur les instances de conteneur du cluster à accéder à d'autres services AWS, notamment Amazon ECS. Lorsque vous créez votre première couche de cluster ECS, choisissez Nouveau profil avec accès ECS pour demander à AWS OpsWorks Stacks de créer le profil requis, qui est nommé`aws-opsworks-ec2-role-with-ecs`. Vous pouvez ensuite utiliser ce profil pour toutes les couches suivantes du cluster ECS. Pour plus d'informations sur le profil d'instance, consultez Spécification des autorisations pour les applications exécutées sur EC2 des instances.

Vous pouvez spécifier d'autres paramètres en modifiant la configuration de la couche, notamment :

- Attache un équilibreur de charge Elastic Load Balancing à la couche.

  Cette approche peut convenir à certains cas d'utilisation, mais Amazon ECS propose des options plus sophistiquées. Pour plus d'informations, consultez Équilibrage de charge des services.
- Spécifier s'il faut attribuer automatiquement des adresses IP publiques ou Elastic aux instances de conteneur.

  Si vous désactivez l'affectation automatique pour les deux types d'adresses, l'instance ne vient pas en ligne, sauf si le sous-réseau dispose d'un NAT configuré correctement. Pour de plus amples informations, veuillez consulter Running a Stack in a VPC.

Gestion du cluster ECS

Après avoir créé une couche de cluster ECS, vous pouvez utiliser AWS OpsWorks Stacks pour gérer le cluster comme suit :

Mettre en service et gérer des instances de conteneur

Au départ, une couche de cluster ECS n'inclut aucune instance de conteneur, même si le cluster d'origine en comprenait. Une option consiste à gérer les instances de la couche à l'aide d'une combinaison appropriée des éléments suivants :

- Ajoutez manuellement des instances 24/7 à la couche et supprimez-les lorsqu'elles ne sont plus nécessaires.
- Ajoutez ou supprimez des instances sur une planification en ajoutant des instances basées sur le temps à la couche.
- Ajoutez ou supprimez des instances en fonction des métriques ou des CloudWatch alarmes de l'hôte AWS OpsWorks Stacks en ajoutant des instances basées sur la charge à la couche.

> **Note**
>
> Si Amazon ECS n'est pas compatible avec le système d'exploitation par défaut de la pile, vous devez spécifier explicitement un système d'exploitation compatible : Amazon Linux 2, Amazon Linux 2018.03, 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 ou Custom — Lorsque vous créez les instances de conteneur. N'utilisez pas l'AMI optimisée ECS pour créer des instances dans une couche ECS, car cette AMI inclut déjà l'agent ECS. AWS OpsWorks Stacks tente également d'installer l'agent ECS pendant le processus de configuration de l'instance, et le conflit peut entraîner l'échec de l'installation.

Pour plus d'informations, consultezOptimisation du nombre de serveurs. AWS OpsWorks Stacks attribue le groupe de sécurité AWS- OpsWorks -ECS-Cluster à chaque instance. Une fois le démarrage de chaque nouvelle instance terminé, AWS OpsWorks Stacks la convertit en instance de conteneur en installant Docker et l'agent Amazon ECS, puis en enregistrant l'instance auprès du cluster.

Si vous préférez utiliser des instances de conteneur existantes, vous pouvez les enregistrer dans la pile et les affecter à la couche ECS Cluster. Veuillez noter que les instances doivent exécuter un système d'exploitation pris en charge, Amazon Linux 2015.03 ou une version ultérieure, ou Ubuntu 14.04 LTS ou une version ultérieure.

> **Note**
>
> Une instance de conteneur ne peut pas appartenir à la fois à une couche de cluster ECS et à une autre couche intégrée. Toutefois, une instance de conteneur peut appartenir à une couche de cluster ECS et à une ou plusieurs couches personnalisées.

Exécuter les mises à jour du système d'exploitation et des packages

Une fois le démarrage d'une nouvelle instance terminé, AWS OpsWorks Stacks installe les dernières mises à jour. Vous pouvez ensuite utiliser AWS OpsWorks Stacks pour maintenir les instances de conteneur à jour. Pour de plus amples informations, veuillez consulter Gestion des mises à jour de sécurité.

Gérer les autorisations utilisateur

AWS OpsWorks Stacks fournit un moyen simple de gérer les autorisations sur les instances de conteneur, y compris la gestion des clés SSH des utilisateurs. Pour plus d'informations, consultez Gestion des autorisations utilisateur et Gestion de l'accès SSH.

Surveiller les métriques de performances

AWS OpsWorks Stacks propose différentes méthodes pour surveiller les indicateurs de performance de la pile, de la couche ou des instances individuelles. Pour de plus amples informations, veuillez consulter Surveillance.

Vous gérez d'autres tâches de gestion, telles que la création de tâches ou de services, via Amazon ECS. Pour plus d'informations, consultez le Guide du développeur Amazon Elastic Container Service.

> **Note**
>
> Pour accéder directement à la page du cluster sur la console Amazon ECS, choisissez Instances, puis choisissez ECS Cluster, qui se trouve dans le coin supérieur droit de la section de la couche de cluster ECS.

Supprimer une couche de cluster ECS d'une pile

Lorsque vous n'avez plus besoin du cluster, supprimez la couche de cluster ECS et annulez l'enregistrement du cluster associé. La suppression d'un cluster sur une pile nécessite deux opérations : l'annulation de l'enregistrement du cluster, puis la suppression de la couche associée. La console AWS OpsWorks Stacks combine ces étapes ; la suppression de couches annule automatiquement l'enregistrement du cluster spécifié. Si vous utilisez l'API, la CLI ou le SDK AWS OpsWorks Stacks, vous devez effectuer des opérations distinctes pour désenregistrer le cluster et supprimer la couche associée.

Pour utiliser la console pour supprimer une couche de cluster ECS

1. Si vous souhaitez contrôler la façon dont les tâches sont arrêtées, utilisez la console, l'API ou la CLI Amazon ECS pour réduire et supprimer les services du cluster. Pour plus d'informations, consultez Nettoyage de vos ressources Amazon ECS.
2. Arrêtez les instances de la couche, puis supprimez-les. Lorsque vous arrêtez une instance de conteneur, AWS OpsWorks Stacks arrête automatiquement toutes les tâches en cours d'exécution, désenregistre l'instance du cluster et met fin à l'instance.

   > **Note**
   >
   > Si vous avez enregistré les instances de conteneur existantes auprès de la pile, vous pouvez annuler l'affectation des instances de la couche, puis annuler leur enregistrement, ce qui redonne le contrôle des instances à ECS.

3. Supprimez la couche. AWS OpsWorks Stacks annule l'enregistrement du cluster associé, mais n'est pas supprimé. Le cluster reste dans Amazon ECS.

Couches AWS OpsWorks de piles personnalisées

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une couche personnalisée possède uniquement un ensemble minimal de recettes. Vous ajoutez ensuite les fonctionnalités appropriées à la couche en implémentant les recettes personnalisées et en les attribuant aux événements de cycle de vie de la couche.

La couche personnalisée a les paramètres de configuration suivants.

> **Note**
>
> AWS OpsWorks Stacks installe automatiquement Ruby sur les instances de la couche. Si vous voulez exécuter le code Ruby sur l'instance, mais que vous ne voulez pas utiliser la version Ruby par défaut, vous pouvez utiliser un fichier JSON personnalisé ou un fichier d'attributs personnalisé pour spécifier votre version préférée. Pour de plus amples informations, veuillez consulter Versions de Ruby.

La procédure de base pour créer une couche personnalisée se compose des étapes suivantes :

1. Implémentez un livre de recettes qui contient les recettes et les fichiers associés requis pour installer et configurer les packages, gérer les modifications de configuration, déployer les applications, etc.

   En fonction de vos besoins, il se peut que vous ayez aussi besoin de recettes pour gérer l'annulation du déploiement et arrêter les tâches. Pour de plus amples informations, veuillez consulter Livres de recettes et recettes.
2. Créez une couche personnalisée.
3. Attribuez vos recettes aux événements du cycle de vie appropriés.

Puis, vous ajoutez les instances à la couche, les démarrez et y déployez les applications.

> **Important**
>
> Pour déployer des applications sur des instances d'une couche personnalisée, vous devez implémenter des recettes pour gérer l'opération de déploiement et les assigner à l'événement Deploy de la couche.

Installations du package de système d'exploitation par couche

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

A partir de Chef 12, vous devez utiliser des recettes personnalisées pour installer les packages sur des couches qui exécutent différents systèmes d'exploitation. Cette approche vous fournit une flexibilité maximale et un contrôle sur les installations de package.

Supposons, par exemple, que vous souhaitiez installer Apache sur des couches en cours d'exécution RedHat, sur les versions Ubuntu et Amazon du système d'exploitation Linux. Le package Apache pour RedHat Amazon Linux est appelé`httpd`, mais sur Ubuntu, il s'appelle`apache2`.

Pour gérer la différence de nom du package, vous pouvez utiliser une syntaxe similaire à celle de l'exemple de recette suivant. La recette installe le package Apache approprié pour chaque système d'exploitation. Cet exemple est basé sur la documentation de Chef.

```
package "Install Apache" do
   case node[:platform]
      when "redhat", "amazon"
         package_name "httpd"
      when "ubuntu"
         package_name "apache2"
   end
end
```

Pour plus d'informations sur l'utilisation de la ressource `package` pour gérer les packages, accédez à la page package dans la documentation de Chef.

Vous pouvez aussi utiliser la méthode d'assistance `value_for_platform` du langage spécifique à un domaine de la recette Chef qui effectue la même chose plus brièvement :

```
package "Install Apache" do
   package_name value_for_platform(
      ["redhat", "amazon"] => { "default" => "httpd" },
      ["ubuntu"] => { "default" => "apache2" }
   )
end
```

Pour plus d'informations sur l'utilisation de la méthode d'assistance `value_for_platform`, consultezA propos du langage spécifique à un domaine pour la recette.

instances

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une instance représente une ressource informatique, telle qu'une EC2 instance Amazon, qui gère le service des applications, l'équilibrage du trafic, etc. Le système d'exploitation d'une instance peut avoir plusieurs distributions Linux ou Windows Server 2012 R2.

Vous pouvez ajouter des instances à une pile de l'une des façons suivantes :

- Utilisez AWS OpsWorks Stacks pour ajouter des instances à une pile. Les instances que vous ajoutez représentent des EC2 instances Amazon.
- Pour les stacks basés sur Linux, vous pouvez enregistrer des instances créées ailleurs, y compris des instances que vous avez créées avec Amazon EC2 et des instances sur site qui s'exécutent sur votre propre matériel.

  Vous pouvez ensuite utiliser AWS OpsWorks Stacks pour gérer ces instances de la même manière que les instances créées avec AWS OpsWorks Stacks.

Cette section décrit comment utiliser AWS OpsWorks Stacks pour créer et gérer des instances.

Rubriques

- Utilisation des AWS OpsWorks instances Stacks
- Utilisation des ressources de calcul créées en dehors d' AWS OpsWorks Stacks
- Modification de la configuration de l'instance
- Supprimer des AWS OpsWorks instances de Stacks
- Utilisation de SSH pour se connecter à une instance Linux
- Utilisation du protocole RDP pour se connecter à une instance Windows

Utilisation des AWS OpsWorks instances Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez utiliser AWS OpsWorks Stacks pour créer des instances et les ajouter à la pile.

Rubriques

- AWS OpsWorks Systèmes d'exploitation Stacks
- Ajout d'une instance à une couche
- Utilisation de Custom AMIs
- Démarrage, arrêt et redémarrage manuels des instances 24/7
- Gestion de la charge avec des instances basées sur le temps et sur la charge

AWS OpsWorks Systèmes d'exploitation Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks prend en charge les versions 64 bits de plusieurs systèmes d'exploitation intégrés, notamment les distributions Linux Amazon et Ubuntu, et Microsoft Windows Server. Quelques commentaires généraux :

- Les instances d'une pile peuvent exécuter Linux ou Windows.

Une pile peut avoir des versions ou des distributions différentes de Linux sur des instances différentes, mais vous ne pouvez pas combiner les instances Linux et Windows.

- Vous pouvez utiliser des images personnalisées AMIs (Amazon Machine Images), mais elles doivent être basées sur l'un des modèles AWS OpsWorks compatibles avec Stacks décrits dans les rubriques de AMIs cette section. Bien qu'il soit possible de créer ou d'enregistrer des instances avec d'autres systèmes d'exploitation (tels que CentOS 6). x) qui ont été créés à partir de données personnalisées ou générées par la communauté AMIs, ils ne sont pas officiellement pris en charge.
  - Systèmes d'exploitation Linux
  - Microsoft Windows Server
- Vous pouvez démarrer et arrêter les instances manuellement ou demander à AWS OpsWorks Stacks de dimensionner automatiquement le nombre d'instances.

  Vous pouvez utiliser le dimensionnement automatique basé sur le temps avec n'importe quelle pile ; les piles Linux peuvent également utiliser le dimensionnement basé sur la charge.
- En plus d'utiliser AWS OpsWorks Stacks pour créer des EC2 instances Amazon, vous pouvez également enregistrer des instances auprès d'une pile Linux qui ont été créées en dehors de AWS OpsWorks Stacks.

  Cela inclut EC2 les instances Amazon et les instances exécutées sur votre propre matériel. Cependant, elles doivent être exécutées sur l'une des distributions Linux prises en charge. Vous ne pouvez pas enregistrer d'instances Amazon EC2 ou Windows sur site.

Vous pouvez exécuter l'`DescribeOperatingSystems`API AWS OpsWorks Stacks pour renvoyer une liste des systèmes d'exploitation pris en charge et de leurs versions prises en charge de Chef. Voici un exemple de commande via AWS CLI :

```
aws opsworks describe-operating-systems
```

Voici un exemple de réponse.

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

```
}
```

Rubriques

- Systèmes d'exploitation Linux
- Microsoft Windows Server

Systèmes d'exploitation Linux

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks prend en charge les versions 64 bits des systèmes d'exploitation Linux suivants.

- Amazon Linux et Amazon Linux 2 (consultez la console AWS OpsWorks Stacks pour les versions actuellement prises en charge)
- Ubuntu 20.04 LTS
- CentOS 7
- Red Hat Enterprise Linux 7

Vous pouvez également utiliser la personnalisation AMIs en fonction de ces systèmes d'exploitation.

Quelques notes générales sur les instances Linux :

Versions de package prises en charge

Les versions prises en charge et les niveaux de correctifs pour les packages, comme Ruby, dépendent du système d'exploitation et la version, comme décrit dans les sections suivantes.

Mises à jour

Par défaut, AWS OpsWorks Stacks garantit que les instances Linux disposent des derniers correctifs de sécurité en appelant automatiquement `yum update` ou `apt-get update` après le démarrage d'une instance. Pour désactiver les mises à jour automatiques, utilisez les UpdateLayeractions CreateInstanceUpdateInstance, CreateLayer, ou, ou les méthodes du SDK AWS ou les commandes de l'interface de ligne de commande AWS équivalentes, pour définir le `InstallUpdatesOnBoot` paramètre sur. `false`

Pour éviter les interruptions de service, AWS OpsWorks Stacks n'installe pas automatiquement les mises à jour une fois qu'une instance est en ligne. Vous pouvez mettre à jour manuellement le système d'exploitation d'une instance en ligne à tout moment en exécutant la commande de pile Upgrade Operating System. Pour plus d'informations sur la gestion des mises à jour de sécurité, consultez Gestion des mises à jour de sécurité.

Pour mieux contrôler la façon dont AWS OpsWorks Stacks met à jour vos instances, créez une AMI personnalisée basée sur l'un des systèmes d'exploitation pris en charge. Par exemple, avec custom, AMIs vous pouvez spécifier les versions de package installées sur une instance. Chaque distribution Linux a des chronologies de prise en charge et des stratégies de fusion de packages différentes, donc vous devez choisir l'approche la plus adaptée à vos besoins. Pour de plus amples informations, veuillez consulter Utilisation de Custom AMIs.

Fichier d'hôtes

Chaque instance en ligne possède un `/etc/hosts` fichier qui associe les adresses IP aux noms d'hôtes. AWS OpsWorks Stacks inclut les adresses publiques et privées de toutes les instances en ligne de la pile dans le `hosts` fichier de chaque instance. Supposons, par exemple, que vous disposiez d'une pile contenant deux instances de Node.js App Server, nodejs-app1 et nodejs-app2, et une instance MySQL, db-master1. Le fichier `hosts` de l'instance nodejs-app1 ressemblera à quelque chose de similaire à l'exemple suivant et les autres instances auront des fichiers `hosts` similaires.

```
...
# 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 Support de proxy pour les agents Stacks

L'agent AWS OpsWorks Stacks pour Chef 11.10 et versions ultérieures inclut un support de base pour les serveurs proxy, qui sont généralement utilisés avec des serveurs isolés. VPCs Pour activer la prise en charge du serveur proxy, une instance doit avoir un fichier `/etc/environment` qui fournit les paramètres appropriés pour le trafic HTTP et HTTPS. Le fichier doit être similaire à ce qui suit, en sachant que vous remplacez le texte en surbrillance par l'URL et un port de votre serveur proxy :

```
http_proxy="http://myproxy.example.com:8080/"
https_proxy="http://myproxy.example.com:8080/"
no_proxy="169.254.169.254"
```

Pour activer la prise en charge de proxy, nous vous recommandons de créer une AMI personnalisée qui inclut un fichier `/etc/environment` approprié et qui utilise cette AMI pour créer vos instances.

> **Note**
>
> Nous vous déconseillons d'utiliser une recette personnalisée pour créer un `/etc/environment` fichier sur vos instances. AWS OpsWorks Stacks a besoin des données du serveur proxy au début du processus de configuration, avant l'exécution de toute recette personnalisée.

Rubriques

- Amazon Linux
- Ubuntu LTS
- CentOS
- Utilisation de Red Hat Enterprise Linux

Amazon Linux

AWS OpsWorks Stacks prend en charge les versions 64 bits d'Amazon Linux et Amazon Linux 2. En plus des mises à jour et des correctifs réguliers, Amazon Linux lance une nouvelle version

environ tous les six mois, ce qui peut impliquer des modifications importantes. Lorsque vous créez une pile ou une instance, vous devez spécifier la version d'Amazon Linux à utiliser. Lorsqu'AWS lance une nouvelle version, vos instances continuent à exécuter la version spécifiée jusqu'à ce que vous la remplaciez explicitement. Une fois qu'une nouvelle version d'Amazon Linux a été lancée, il y a une période de quatre semaines de migration au cours de laquelle AWS continue à fournir des mises à jour régulières pour l'ancienne version. Après la fin de la période de migration, vos instances peuvent continuer à exécuter l'ancienne version, mais AWS ne fournit plus de mises à jour. Pour plus d'informations, consultez AMI Amazon Linux FAQs.

Lorsqu'une nouvelle version d'Amazon Linux est disponible, nous vous recommandons de mettre à jour vers la nouvelle version pendant la période de migration afin que vos instances continuent de recevoir des mises à jour de sécurité. Avant de mettre à jour les instances de votre pile de production, nous vous recommandons de commencer une nouvelle instance et de vérifier que votre application est exécutée correctement sur la nouvelle version. Vous pouvez ensuite mettre à jour les instances de la pile de production.

> **Note**
>
> Par défaut, les versions personnalisées AMIs basées sur Amazon Linux sont automatiquement mises à jour vers la nouvelle version lors de leur publication. La méthode recommandée consiste à verrouiller votre AMI personnalisée sur une version Amazon Linux spécifique, ce qui vous permet de reporter la mise à jour jusqu'à ce que vous ayez testé la nouvelle version. Pour plus d'informations, consultez Comment verrouiller mon AMI avec une version spécifique ?.
>
> Si vous utilisez un AWS CloudFormation modèle pour créer des piles avec des instances exécutant Amazon Linux, les modèles doivent spécifier explicitement une version d'Amazon Linux. En particulier, si votre modèle spécifie `Amazon Linux`, les instances continueront d'exécuter la version 2016.09. Pour plus d'informations, consultez AWS::OpsWorks::Stack et AWS::OpsWorks::Instance.

Pour mettre à jour la version Amazon Linux d'une instance, effectuez l'une des actions suivantes :

- Pour les instances en ligne, exécutez la commande de pile Upgrade Operating System.

  Lorsqu'une nouvelle version d'Amazon Linux est disponible, les pages Instances et Stack affichent un avis avec un lien qui vous conduit jusqu'à la page Run Command (Exécuter une

commande). Vous pouvez ensuite exécuter Upgrade Operating System (Mettre à niveau le système d'exploitation) pour mettre à niveau votre instance.

- Pour les instances hors ligne soutenues par Amazon Elastic Block Store (soutenues par EBS), démarrez les instances et exécutez Upgrade Operating System, comme décrit dans l'instruction précédente.
- Pour les instances basées sur le stockage d'instance hors connexion, y compris les instances basées sur le temps et sur la charge, modifiez le paramètre Operating system (Système d'exploitation) de l'instance pour spécifier la nouvelle version.

  AWS OpsWorks Stacks met automatiquement à jour les instances vers la nouvelle version lorsqu'elles sont redémarrées.

Amazon Linux : versions de Node.js prises en charge

Version d'Amazon Linux	Versions Node.js
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

Version d'Amazon Linux	Versions Node.js
	0.12.13 0.12.12 0.12.10

Amazon Linux : versions de Chef prises en charge

Version de Chef	Versions d'Amazon Linux prises en charge
12	Amazon Linux 2 Amazon Linux 2018.03 Amazon Linux 2017.09 Amazon Linux 2017.03 Amazon Linux 2016.09 Amazon Linux 2016.03
11.10	Amazon Linux 2018.03 Amazon Linux 2017.09 Amazon Linux 2017.03 Amazon Linux 2016.09 Amazon Linux 2016.03
11.4 (deprecated)	Amazon Linux 2016.09 Amazon Linux 2016.03

Important

Avant de mettre à jour les instances t1.micro, assurez-vous qu'elles ont un fichier d'échange temporaire, `/var/swapfile`. Les instances t1.micro sur les piles de Chef 0.9 n'ont pas de fichier d'échange. Pour les piles de Chef 11.4 et Chef 11.10, les versions récentes de l'agent d'instance créent automatiquement un fichier d'échange pour les instances t1.micro. Toutefois, ce changement a été effectué sur une période de plusieurs semaines, donc vous devez vérifier l'existence de `/var/swapfile` sur les instances créées avant le 24 mars 2014 environ.

Pour les instances t1.micro qui n'ont pas de fichier d'échange, vous pouvez en créer un comme suit :

- Pour la version Chef 11.10 et les versions ultérieures, créez de nouvelles instances t1.micro qui ont automatiquement un fichier d'échange.
- Pour les piles de Chef 0.9, exécutez les commandes suivantes sur chaque instance en tant qu'utilisateur racine.

```
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
```

Vous pouvez également utiliser ces commandes sur la version Chef 11.10 et les piles ultérieures si vous ne souhaitez pas créer de nouvelles instances.

Ubuntu LTS

Ubuntu lance une nouvelle version d'Ubuntu LTS environ tous les deux ans et prend en charge chaque version pendant environ cinq ans. Ubuntu fournit des mises à jour et des correctifs de sécurité pendant la durée de la prise en charge du système d'exploitation. Pour plus d'informations, consultez LTS - Ubuntu Wiki.

- Vous ne pouvez pas mettre à jour une instance Ubuntu existante vers une version plus récente d'Ubuntu.

  Vous devez créer une nouvelle instance Ubuntu et supprimer l'ancienne instance.
- Ubuntu 20.04 LTS n'est pris en charge que pour les piles Chef 12 et supérieures.

CentOS

AWS OpsWorks Stacks prend en charge la version 64 bits de CentOS 7. La version initialement prise en charge est CentOS 7 et CentOS publie une nouvelle version tous les deux ans environ.

Lorsque vous démarrez une nouvelle instance dans une pile CentOS, AWS OpsWorks Stacks installe automatiquement la version la plus récente de CentOS. Comme AWS OpsWorks Stacks ne met pas automatiquement à jour le système d'exploitation des instances existantes lorsqu'une nouvelle

version mineure de CentOS est publiée, une instance nouvellement créée peut recevoir une version plus récente que les instances existantes de la pile. Pour que toutes les versions soient cohérentes sur votre pile, vous pouvez mettre à jour vos instances existantes vers la version actuelle de CentOS, comme suit :

- Pour les instances en ligne, exécutez la commande de pile Upgrade Operating System (Mettre à niveau le système d'exploitation) qui exécute `yum update` sur les instances spécifiées afin de les mettre à jour vers la version actuelle.

  Lorsqu'une nouvelle version mineure de CentOS 7 est disponible, les pages Instances et Stack (Pile) affichent un avis avec un lien qui vous conduit jusqu'à la page Run Command (Exécuter une commande). Vous pouvez ensuite exécuter Upgrade Operating System (Mettre à niveau le système d'exploitation) pour mettre à niveau vos instances.
- Pour les instances hors ligne basées sur Amazon EBS, démarrez les instances et exécutez Upgrade Operating System comme décrit dans l'élément de liste précédent.
- Pour les instances hors ligne sauvegardées en magasin, AWS OpsWorks Stacks installe automatiquement la nouvelle version au redémarrage des instances.

CentOS : versions de Chef prises en charge

Version de Chef	Version de CentOS prise en charge
`12`	`CentOS 7`
`11.10`	`(None supported)`
`11.4 (deprecated)`	`(None supported)`

> **Note**
>
> AWS OpsWorks Stacks prend en charge Apache 2.4 pour les instances CentOS.

Utilisation de Red Hat Enterprise Linux

AWS OpsWorks Stacks prend en charge la version 64 bits de Red Hat Enterprise Linux 7 (RHEL 7). La version initiale prise en charge est RHEL 7.1 et Red Hat lance une nouvelle version mineure tous les 9 mois environ. Les versions mineures sont généralement compatibles avec RHEL 7.0. Pour plus d'informations, consultez Stratégies de cycle de vie et de mise à jour.

Lorsque vous démarrez une nouvelle instance, AWS OpsWorks Stacks installe automatiquement la version actuelle de RHEL 7. Comme AWS OpsWorks Stacks ne met pas automatiquement à jour le système d'exploitation des instances existantes lorsqu'une nouvelle version mineure de RHEL 7 est publiée, une instance nouvellement créée peut recevoir une version plus récente que les instances existantes de la pile. Pour que toutes les versions soient cohérentes sur votre pile, vous pouvez mettre à jour vos instances existantes vers la version actuelle de RHEL 7, comme suit :

- Pour les instances en ligne, exécutez la commande de pile Upgrade Operating System (Mettre à niveau le système d'exploitation) qui exécute `yum update` sur les instances spécifiées afin de les mettre à jour vers la version actuelle.

  Lorsqu'une nouvelle version mineure de RHEL 7 est disponible, les pages Instances et Stack (Pile) affichent un avis avec un lien qui vous conduit jusqu'à la page Run Command (Exécuter une commande). Vous pouvez ensuite exécuter Upgrade Operating System (Mettre à niveau le système d'exploitation) pour mettre à niveau vos instances.
- Pour les instances hors ligne basées sur Amazon EBS, démarrez les instances et exécutez Upgrade Operating System comme décrit dans l'élément de liste précédent.
- Pour les instances hors ligne sauvegardées en magasin, AWS OpsWorks Stacks installe automatiquement la nouvelle version au redémarrage des instances.

Red Hat Enterprise Linux : versions Node.js prises en charge

Version de l'RHEL	Versions Node.js
7	(Node.js versions only apply to Chef 11.10 stacks) 0.8.19 0.8.26 0.10.11 0.10.21 0.10.24 0.10.25

Version de l'RHEL	Versions Node.js
	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 : versions prises en charge de Chef

Version de Chef	Versions d'RHEL prise en charge
12	Red Hat Enterprise Linux 7
11.10	Red Hat Enterprise Linux 7
11.4 (deprecated)	(None supported)

Toutes les versions de Node.js antérieures à 0.10.40 sont obsolètes. Les versions 0.12.7 et 0.12.9 sont également obsolètes.

> **Note**
>
> AWS OpsWorks Stacks prend en charge Apache 2.4 pour les instances de RHEL 7.

Microsoft Windows Server

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les notes suivantes décrivent le support AWS OpsWorks de Stacks pour les instances Windows. Les instances Windows sont disponibles uniquement pour les piles de Chef 12.2. La version exacte de Chef dans une pile Windows est 12.22.

À l'heure actuelle, l'agent AWS OpsWorks Stacks ne peut pas être installé sur des instances Windows qui utilisent une langue d'interface utilisateur système autre que l'anglais - États-Unis d'Amérique (en-US), et AWS OpsWorks Stacks ne peut pas les gérer.

Versions

AWS OpsWorks Stacks prend en charge les versions 64 bits de Windows suivantes :

- Microsoft Windows Server 2022 Base
- Microsoft Windows Server 2022 avec SQL Server Express
- Microsoft Windows Server 2022 avec SQL Server Standard
- Microsoft Windows Server 2022 avec SQL Server Web
- Microsoft Windows Server 2019 Base
- Microsoft Windows Server 2019 avec SQL Server Express
- Microsoft Windows Server 2019 avec SQL Server Standard
- Microsoft Windows Server 2019 avec SQL Server Web

Création d'instances

Vous créez des instances Windows à l'aide de la console, de l'API ou de la CLI AWS OpsWorks Stacks. Les instances Windows sont soutenues par Amazon EBS, mais vous ne pouvez pas monter de volumes Amazon EBS supplémentaires.

Les piles Windows peuvent utiliser des instances 24/7 que vous arrêtez et démarrez manuellement. Elles peuvent également utiliser la scalabilité automatique basée sur le temps, qui démarre et arrête automatiquement les instances selon sur une planification spécifiée par l'utilisateur. Les piles basées sur Windows ne peuvent pas utiliser la scalabilité automatique basée sur la charge.

Vous ne pouvez pas enregistrer des instances Windows créées en dehors de AWS OpsWorks Stacks avec une pile.

Mises à jour

AWS met à jour Windows AMIs pour chaque ensemble de correctifs. Ainsi, lorsque vous créez une instance, elle dispose des dernières mises à jour. Cependant, AWS OpsWorks Stacks ne fournit aucun moyen d'appliquer des mises à jour aux instances Windows en ligne. La façon la plus simple pour s'assurer que Windows est à jour consiste à remplacer vos instances régulièrement, afin qu'elles exécutent toujours l'AMI la plus récente.

Couches

Pour gérer des tâches telles que l'installation et la configuration du logiciel ou le déploiement des applications, vous devez implémenter une ou plusieurs couches personnalisées avec des recettes personnalisées.

Chef

Les instances Windows utilisent Chef 12.22 et exécutent le client Chef en mode local, ce qui lance un serveur Chef en mémoire local nommé chef-zero. La présence de ce serveur permet aux recettes personnalisées d'utiliser les conteneurs de données et la recherche Chef.

Connexion à distance

AWS OpsWorks Stacks fournit aux utilisateurs IAM autorisés un mot de passe qu'ils peuvent utiliser pour se connecter aux instances Windows. Ce mot de passe expire après une durée spécifiée. Les administrateurs peuvent utiliser une paire de clés SSH pour récupérer le mot de passe administrateur d'une instance, ce qui leur fournit un accès RDP illimité. Pour de plus amples informations, veuillez consulter Connexion avec RDP.

Kit SDK AWS

AWS OpsWorks Stacks les installe automatiquement AWS SDK pour .NETsur chaque instance. Ce package inclut les bibliothèques AWS .NET et les outils AWS pour Windows, y compris les outils AWS pour PowerShell. Pour utiliser le kit SDK Ruby, vous pouvez demander à une recette personnalisée d'installer la GEM appropriée.

Surveillance et mesures

Les instances Windows prennent en charge les métriques Amazon CloudWatch (CloudWatch) standard, que vous pouvez consulter dans la CloudWatch console.

Ruby

Le client Chef 12.22 que AWS OpsWorks Stacks installe sur les instances Windows est livré avec Ruby 2.3.6. Cependant, AWS OpsWorks Stacks n'ajoute pas le répertoire de l'exécutable à la

variable d'environnement PATH. Pour que vos applications utilisent cette version de Ruby, vous la trouverez généralement dans `C:\opscode\chef\embedded\bin\`.

AWS OpsWorks CLI de l'agent Stacks

L'agent AWS OpsWorks Stacks sur les instances Windows n'expose pas d'interface de ligne de commande.

Support de proxy

Procédez comme suit pour configurer la prise en charge de proxy pour les instances Windows :

1. Modifiez `machine.config` pour ajouter ce qui suit, qui ajoute la prise en charge du proxy aux PowerShell applications Windows (bootstrap initial) et .NET (agent AWS OpsWorks Stacks) :

```
<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. Exécutez les commandes suivantes pour définir des variables d'environnement en vue d'une utilisation ultérieure par Chef et Git :

```
setx /m no_proxy "localhost,169.254.169.254"
setx /m http_proxy "http://10.100.1.91:3128"
setx /m https_proxy "http://10.100.1.91:3128"
```

> **Note**
>
> Pour mieux contrôler la façon dont AWS OpsWorks Stacks met à jour vos instances, créez une AMI personnalisée basée sur Microsoft Windows Server 2022 Base. Par exemple, avec custom, AMIs vous pouvez spécifier quel logiciel est installé sur une instance, telle que le

serveur Web (IIS). Pour de plus amples informations, veuillez consulter Utilisation de Custom AMIs.

Ajout d'une instance à une couche

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une fois que vous avez créé une couche, vous ajoutez généralement au moins une instance. Vous pouvez ajouter d'autres instances plus tard, si l'ensemble actuel ne peut pas gérer la charge. Vous pouvez aussi utiliser les instances à date définie ou à charge définie pour dimensionner automatiquement le nombre d'instances.

Vous pouvez ajouter des instances nouvelles ou existantes à une couche :

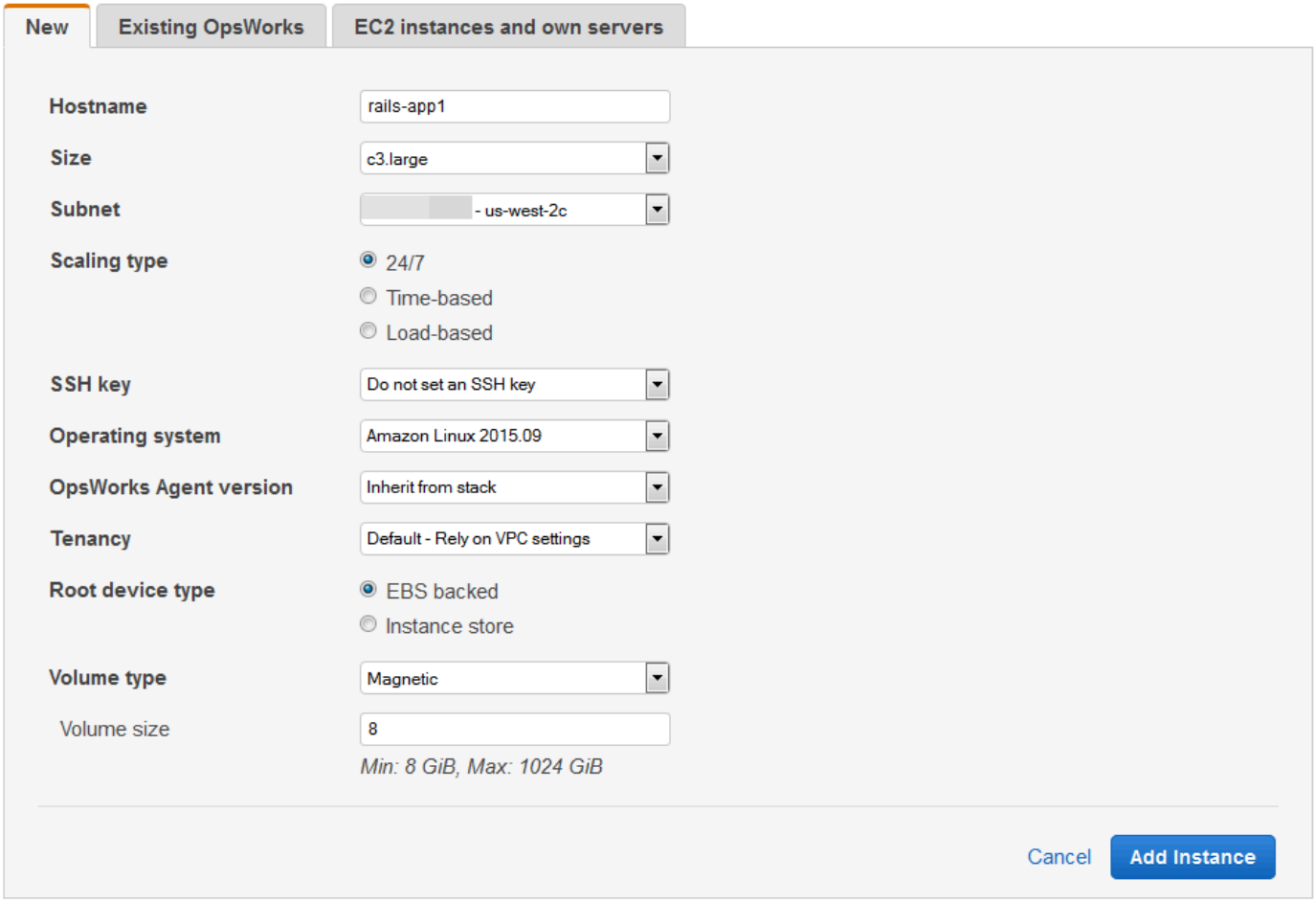
- Nouveau : OpsWorks crée une nouvelle instance, configurée selon vos spécifications, et en fait un membre de la couche.
- Existant —Vous pouvez ajouter une instance existante à partir de n'importe quelle couche compatible, mais elle doit être à l'état hors ligne (arrêtée).

Si une instance appartient à plusieurs couches, AWS OpsWorks Stacks exécute les recettes de chacune des couches de l'instance quand un événement de cycle de vie se produit, ou lorsque vous exécutez une commande de pile ou de déploiement.

Vous pouvez également rendre une instance membre de plusieurs couches en modifiant sa configuration. Pour de plus amples informations, veuillez consulter Modification de la configuration de l'instance.

Pour ajouter une nouvelle instance à une couche

1. Sur la page Instances, choisissez +Instance pour la couche appropriée et (si nécessaire) sélectionnez l'onglet New (Nouveau). Si vous souhaitez configurer d'autres options en plus de Host name (Nom d'hôte), Size (Taille) et Subnet (Sous-réseau) ou Availability Zone (Zone de disponibilité), choisissez Advanced >> (Avancé >>) pour afficher plus d'options. L'exemple suivant montre l'ensemble complet des options :



2. Si vous le souhaitez, vous pouvez remplacer la configuration par défaut, dont vous avez spécifié la plus grande part lorsque vous avez créé la pile. Pour de plus amples informations, veuillez consulter Créer une pile.

   Hostname

   Identifie l'instance sur le réseau. Par défaut, AWS OpsWorks Stacks génère le nom d'hôte de chaque instance en utilisant le thème Hostname que vous avez spécifié lors de la création de la pile. Vous pouvez remplacer cette valeur et spécifier votre nom d'hôte préféré.

Size

Type d' EC2 instance Amazon, qui spécifie les ressources de l'instance, telles que la quantité de mémoire ou le nombre de cœurs virtuels. AWS OpsWorks Stacks spécifie une taille par défaut pour chaque instance, que vous pouvez remplacer par votre type d'instance préféré.

Les types d'instances pris en charge par AWS OpsWorks Stacks varient selon que la pile se trouve ou non dans un VPC. Les types d'instance sont également limités si votre compte utilise l'offre gratuite AWS. La liste déroulante Size (Taille) affiche les types d'instance pris en charge pour la version de Chef que votre pile prend en charge. N'oubliez pas que les micro-instances telles que les instances t1.micro peuvent ne pas avoir suffisamment de ressources pour prendre en charge certaines couches. Pour de plus amples informations, veuillez consulter Types d'instance .

> **Note**
>
> Si vous utilisez des instances à charge équilibrée, notez que la configuration des événements de cycle de vie peut produire un pic de charge élevé pour l'UC, susceptible de durer une minute ou plus. Avec les instances plus petites, ce pic de charge peut suffire à déclencher un agrandissement, en particulier pour les piles à charge équilibrée importantes et ayant des événements Configure fréquents. Voici quelques façons de réduire la probabilité qu'un événement Configurer n'entraîne un agrandissement superflu.
>
> - Utilisez des instances plus volumineuses, afin que la charge supplémentaire d'un événement Configure ne suffise pas à déclencher un agrandissement.
> - N'utilisez pas les types d'instance comme T2 qui partagent les ressources UC.
>
> Cela garantit que lorsqu'un événement Configure intervient, toutes les ressources UC de l'instance sont immédiatement disponibles.
> - Rendre `exceeded threshold` significativement plus long que le temps nécessaire pour traiter un événement Configure, peut-être 5 minutes.
>
> Pour de plus amples informations, veuillez consulter Utilisation du dimensionnement automatique basé sur la charge.

Availability Zone/Subnet

Si la pile n'est pas dans un VPC, ce paramètre a pour nom Availability Zone (Zone de disponibilité) et affiche les zones de la région. Vous pouvez utiliser ce paramètre pour remplacer la zone de disponibilité par défaut que vous avez spécifiée lorsque vous avez créé la pile.

Si la pile est en cours d'exécution dans un VPC, ce paramètre a pour nom Subnet (Sous-réseau) et affiche les sous-réseaux du VPC. Vous pouvez utiliser ce paramètre pour remplacer le réseau par défaut que vous avez spécifié lorsque vous avez créé la pile.

> **Note**
>
> Par défaut, AWS OpsWorks Stacks répertorie les plages CIDR du sous-réseau. Pour rendre la liste plus lisible, utilisez la console ou l'API VPC pour ajouter une balise à chaque sous-réseau avec la clé définie sur **`Name`** et la valeur définie sur le nom du sous-réseau. AWS OpsWorks Stacks ajoute ce nom à la plage CIDR. Dans l'exemple précédent, la balise Name (Nom) du sous-réseau est définie sur **`Private`**.

Scaling Type

Détermine comment l'instance est démarrée et arrêtée.

- La valeur par défaut est une instance 24/7, que vous démarrez et arrêtez manuellement.
- AWS OpsWorks Stacks démarre et arrête les instances basées sur le temps selon un calendrier spécifié.
- (Linux uniquement) AWS OpsWorks Stacks démarre et arrête les instances basées sur la charge en fonction des métriques de charge spécifiées.

> **Note**
>
> Vous ne démarrez ni n'arrêtez vous-même les instances à charge définie ou à date définie. Au lieu de cela, vous configurez les instances, et AWS OpsWorks Stacks les démarre et les arrête en fonction de la configuration. Pour de plus amples informations, veuillez consulter Gestion de la charge avec des instances basées sur le temps et sur la charge.

SSH key

Une paire de EC2 clés Amazon. AWS OpsWorks Stacks installe la clé publique sur l'instance.

- Pour les instances Linux, vous pouvez utiliser la clé privée correspondante avec un client SSH pour vous connecter à l'instance.
- Pour les instances Windows, vous pouvez utiliser la clé privée correspondante pour récupérer le mot de passe Administrateur de l'instance. Vous pouvez ensuite utiliser ce mot de passe avec RDP pour vous connecter à l'instance en tant qu'Administrateur.

À l'origine, ce paramètre a la valeur Default SSH key (Clé SSH par défaut) que vous avez spécifiée lorsque vous avez créé la pile.

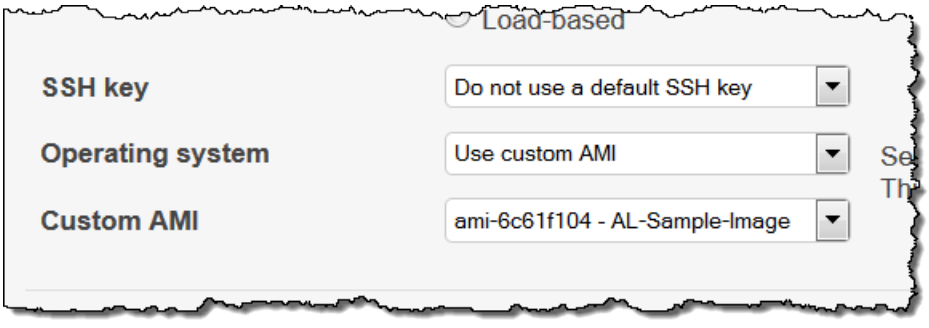
- Si la valeur par défaut est définie sur Ne pas utiliser de clé SSH par défaut, vous pouvez spécifier l'une des EC2 clés Amazon de votre compte.
- Si la valeur par défaut est définie sur une EC2 clé Amazon, vous pouvez spécifier une clé différente ou ne pas spécifier de clé.

Système d'exploitation

Le système d'exploitation indique le système d'exploitation exécuté par l'instance. AWS OpsWorks Stacks ne prend en charge que les systèmes d'exploitation 64 bits.

À l'origine, ce paramètre a la valeur Default operating system (Système d'exploitation par défaut) que vous avez spécifiée lorsque vous avez créé la pile. Vous pouvez remplacer la valeur par défaut pour spécifier un autre système d'exploitation Linux ou une AMI personnalisée. Cependant, vous ne pouvez pas passer de Linux à Windows ou de Windows à Linux.

Si vous sélectionnez Utiliser une AMI personnalisée, la page affiche une liste de périphériques personnalisés AMIs plutôt que d'architecture et de type de périphérique racine.



Pour de plus amples informations, veuillez consulter Utilisation de Custom AMIs.

OpsWorks Version de l'agent

OpsWorks La version de l'agent indique la version de l'agent AWS OpsWorks Stacks que vous souhaitez exécuter sur l'instance. Si vous voulez qu' AWS OpsWorks Stacks mette à jour l'agent automatiquement, choisissez Inherit from stack (Hériter de la pile). Pour installer une version spécifique de l'agent et mettre à jour manuellement l'agent sur l'instance, sélectionnez une version dans la liste déroulante.

> **Note**
>
> Certaines versions de l'agent ne fonctionnent pas avec toutes les versions des systèmes d'exploitation. Si votre instance exécute un agent (ou si vous installez un agent sur une instance) qui n'est pas entièrement pris en charge par le système d'exploitation de l'instance, la console AWS OpsWorks Stacks affiche des messages d'erreur vous demandant d'installer un agent compatible.

Location

Sélectionnez l'option de location pour votre instance. Vous pouvez choisir d'exécuter vos instances sur des serveurs physiques totalement dédiés à votre utilisation.

- Default - Rely on VPC settings (Par défaut - S'appuyer sur les paramètres VPC). Aucune location, ou hérite les paramètres de location de votre VPC.
- Dedicated - Run a dedicated instance (Par défaut - S'appuyer sur les paramètres VPC). Paiement à l'heure pour les instances qui s'exécutent sur un matériel à client unique. Pour plus d'informations, consultez les sections Instances dédiées dans le guide de l'utilisateur Amazon VPC et Instances EC2 dédiées Amazon.
- Dedicated host - Run this instance on a dedicated host (Hôte dédié - Exécuter cette instance sur un hôte dédié). Paiement d'un hôte physique qui est entièrement dédié à l'exécution de vos instances et utilisation du modèle BYOL (Bring-Your-Own-License) pour vos licences logicielles par socket, par cœur ou par ordinateur virtuel afin de réduire les coûts. Pour plus d'informations, consultez la section Présentation des hôtes dédiés dans la EC2 documentation Amazon et Amazon EC2 Dedicated Hosts.

Root device type

Spécifie le stockage du périphérique racine de l'instance.

- Les instances Linux peuvent être soit soutenues par Amazon EBS, soit sauvegardées par des instances stockées.
- Les instances Windows doivent être soutenues par Amazon EBS.

Pour plus d'informations, consultez Stockage.

> **Note**
>
> Après le démarrage initial, les instances basées sur Amazon EBS démarrent plus rapidement que les instances basées sur le stockage d'instance, car AWS OpsWorks Stacks n'a pas à réinstaller le logiciel de l'instance à partir de zéro. Pour de plus amples informations, veuillez consulter Stockage de périphérique racine.

Type de volume

Spécifie le type de volume du périphérique racine : Magnetic (Magnétique), Provisioned IOPS (SSD) [IOPS provisionnées (SSD)] ou General Purpose (SSD) [Usage général (SSD)]. Pour de plus amples informations, veuillez consulter Types de volume Amazon EBS.

Taille du volume

Spécifie la taille du volume du périphérique racine pour le type de volume spécifié. Pour de plus amples informations, veuillez consulter Types de volume Amazon EBS.

- General Purpose (SSD) [Usage général (SSD)]. La taille minimale autorisée est de 8 Gio et la taille maximale de 16384 Gio.
- Provisioned IOPS (SSD) [IOPS provisionnés (SSD)]. La taille minimale autorisée est de 8 Gio et la taille maximale de 16384 Gio. Vous pouvez définir un minimum de 100 opérations d'entrée/sortie par seconde (IOPS) et un maximum de 240 IOPS.
- Magnetic (Magnétique). La taille minimale autorisée est de 8 Gio et la taille maximale de 1024 Gio.

3. Choisissez Add Instance (Ajouter une instance) pour créer la nouvelle instance.

> **Note**
>
> Vous ne pouvez pas remplacer le paramètre de version de l'agent par défaut de la pile lorsque vous créez une instance. Pour spécifier un paramètre de version d'agent personnalisé, vous devez créer l'instance, puis modifier sa configuration.

Pour ajouter une instance existante à une couche

1. Sur la page Instances, choisissez +Instance pour la couche appropriée, puis ouvrez l'onglet Existing (Existante).

   > **Note**
   >
   > Si vous changez d'avis sur l'utilisation d'une instance existante, choisissez New (Nouveau) pour créer une nouvelle instance, comme décrit dans la procédure précédente.

2. Sous l'onglet Existing (Existante), sélectionnez une instance dans la liste.
3. Choisissez Add Instance (Ajouter une instance) pour créer la nouvelle instance.

Une instance représente une EC2 instance Amazon, mais il s'agit essentiellement d'une structure de données AWS OpsWorks Stacks. Une instance doit être démarrée pour créer une EC2 instance Amazon en cours d'exécution, comme décrit dans les sections suivantes.

> **Important**
>
> Si vous lancez des instances dans un VPC par défaut, vous devez être prudent quant à la modification de la configuration du VPC. Les instances doivent toujours être en mesure de communiquer avec le service AWS OpsWorks Stacks, Amazon S3 et les référentiels de packages. Si, par exemple, vous supprimez une passerelle par défaut, les instances perdront leur connexion au service AWS OpsWorks Stacks, qui les traitera alors comme défaillantes et les réparera automatiquement. Cependant, AWS OpsWorks Stacks ne sera pas en mesure d'installer l'agent de l'instance sur les instances réparées. Sans agent, les instances ne peuvent pas communiquer avec le service et le processus de démarrage ne progressera pas au-delà du statut `booting`. Pour plus d'informations sur les VPC par défaut, consultez Plateformes prises en charge.

Vous pouvez également intégrer des ressources informatiques Linux dans une pile créée en dehors de AWS OpsWorks Stacks :

- EC2 Instances Amazon que vous avez créées directement à l'aide de la EC2 console, de la CLI ou de l'API Amazon.
- Instances locales s'exécutant sur votre propre matériel, y compris les instances s'exécutant dans les machines virtuelles.

Pour de plus amples informations, veuillez consulter Utilisation des ressources de calcul créées en dehors d' AWS OpsWorks Stacks.

Utilisation de Custom AMIs

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks propose deux méthodes de personnalisation des instances : Amazon Machine Images (AMIs) personnalisées et recettes Chef. Les deux approches vous permettent de contrôler quels sont les packages et versions de package installés, comment ils sont configurés, et ainsi de suite. Cependant, chaque approche ayant différents avantages, la meilleure dépend de vos besoins.

Les principales raisons à prendre en compte pour l'utilisation d'une AMI personnalisée sont les suivantes :

- Vous voulez regrouper préalablement des packages spécifiques au lieu de les installer une fois que l'instance a démarré.
- Vous voulez contrôler le moment des mises à jour des packages pour fournir une image de base cohérente à votre couche.
- Vous voulez que les instances (notamment celles à charge définie) démarrent aussi rapidement que possible.

Les principales raisons à prendre en compte pour l'utilisation des recettes Chef sont les suivantes :

- Ils sont plus flexibles que les modèles personnalisés AMIs.
- Elles sont plus faciles à mettre à jour.
- Elles peuvent effectuer des mises à jour sur des instances en cours d'exécution.

En pratique, la solution optimale peut être une combinaison de ces deux approches. Pour plus d'informations sur les recettes, consultez Livres de recettes et recettes.

Rubriques

- Comment AMIs fonctionne Custom avec AWS OpsWorks Stacks
- Création d'une AMI personnalisée pour AWS OpsWorks Stacks

Comment AMIs fonctionne Custom avec AWS OpsWorks Stacks

Pour spécifier une AMI personnalisée pour vos instances, sélectionnez Utiliser une AMI personnalisée comme système d'exploitation de l'instance lorsque vous créez une nouvelle instance. AWS OpsWorks Stacks affiche ensuite une liste des éléments personnalisés AMIs dans la région de la pile et vous sélectionnez celui qui convient dans la liste. Pour de plus amples informations, veuillez consulter Ajout d'une instance à une couche.

> **Note**
>
> Vous ne pouvez pas spécifier une AMI personnalisée particulier comme système d'exploitation par défaut d'une pile. Vous pouvez définir `Use custom AMI` comme système d'exploitation par défaut de la pile, mais vous ne pouvez spécifier une AMI particulière que lorsque vous ajoutez de nouvelles instances aux couches. Pour plus d'informations, consultez Ajout d'une instance à une couche et Créer une pile. Bien qu'il soit possible de créer des instances avec d'autres systèmes d'exploitation (tels que CentOS 6). x) qui ont été créés à partir de données personnalisées ou générées par la communauté AMIs, ils ne sont pas officiellement pris en charge.

Cette rubrique présente quelques questions générales que vous devez prendre en compte avant de créer ou d'utiliser une AMI personnalisée.

Rubriques

- Comportement au démarrage
- Choix d'une couche
- Gestion des applications

Comportement au démarrage

Lorsque vous démarrez l'instance, AWS OpsWorks Stacks utilise l'AMI personnalisée spécifiée pour lancer une nouvelle EC2 instance Amazon. AWS OpsWorks Stacks utilise ensuite cloud-init pour installer l'agent AWS OpsWorks Stacks sur l'instance et l'agent exécute les recettes de configuration de l'instance, suivies des recettes de déploiement. Une fois que l'instance est en ligne, l'agent exécute les recettes Configure pour chaque instance de la pile, y compris l'instance nouvellement ajoutée.

Choix d'une couche

L'agent AWS OpsWorks Stacks n'entre généralement pas en conflit avec les packages installés. Toutefois, l'instance doit être membre d'au moins une couche. AWS OpsWorks Stacks exécute toujours les recettes de cette couche, ce qui peut poser problème. Vous devez bien comprendre ce que font les recettes d'une couche avant d'ajouter à celle-ci une instance avec une AMI personnalisée.

Pour voir quelles recettes un type de couche particulier exécute sur votre instance, ouvrez une pile qui inclut cette couche. Puis, cliquez sur Layers (Couches) dans le panneau de navigation, puis sur Recipes (Recettes) pour la couche qui vous intéresse. Pour voir le code, cliquez sur le nom de la recette.

> **Note**
>
> Pour Linux AMIs, l'un des moyens de réduire les risques de conflits consiste à utiliser AWS OpsWorks Stacks pour approvisionner et configurer l'instance qui constitue la base de votre AMI personnalisée. Pour de plus amples informations, veuillez consulter Création d'une AMI Linux personnalisée à partir d'une AWS OpsWorks instance Stacks.

Gestion des applications

En plus des packages, il se peut que vous souhaitiez également inclure une application dans l'AMI. Si vous avez une application complexe, son inclusion dans l'AMI peut réduire la durée de démarrage

de l'instance. Vous pouvez inclure de petites applications dans votre AMI, mais il n'y a généralement que peu ou pas d'avantage en termes de temps par rapport au déploiement de l'application par AWS OpsWorks Stacks.

Une option consiste à inclure l'application dans votre AMI et également à créer une application qui déploie l'application sur les instances à partir d'un référentiel. Cette approche réduit votre temps de démarrage, mais fournit également un moyen pratique de mettre à jour l'application une fois que l'instance est en cours d'exécution. Notez que, comme les recettes Chef sont idempotentes, les recettes du déploiement ne modifient pas l'application aussi longtemps que la version du référentiel est identique à celle de l'instance.

Création d'une AMI personnalisée pour AWS OpsWorks Stacks

Pour utiliser une AMI personnalisée avec AWS OpsWorks Stacks, vous devez d'abord créer une AMI à partir d'une instance personnalisée. Vous avez le choix entre deux options :

- Utilisez la EC2 console ou l'API Amazon pour créer et personnaliser une instance, sur la base d'une version 64 bits de l'une des instances prises en charge par AWS OpsWorks Stacks AMIs.
- Pour Linux AMIs, utilisez OpsWorks pour créer une EC2 instance Amazon, en fonction de la configuration des couches associées.

Avant de créer une AMI Linux personnalisée, désactivez-la `noexec` sur la `/tmp` partition pour permettre à AWS OpsWorks Stacks d'installer son agent sur des instances Linux personnalisées.

> **Note**
>
> Sachez que, comme une AMI peut ne pas fonctionner avec tous les types d'instance, assurez-vous que votre AMI initiale est compatible avec les types d'instance que vous prévoyez d'utiliser. En particulier, les types d'instances R3 nécessitent une AMI de virtualisation à assistance matérielle (HVM).

Vous utilisez ensuite la EC2 console ou l'API Amazon pour créer une AMI personnalisée à partir de l'instance personnalisée. Vous pouvez utiliser votre option personnalisée AMIs dans n'importe quelle pile située dans la même région en ajoutant une instance à une couche et en spécifiant votre AMI personnalisée. Pour plus d'informations sur la création d'une instance qui utilise une AMI personnalisée, consultez Ajout d'une instance à une couche.

> **Note**
>
> Par défaut, AWS OpsWorks Stacks installe toutes les mises à jour d'Amazon Linux au démarrage, ce qui vous fournit la dernière version. De plus, Amazon Linux libère une nouvelle version environ tous les six mois, ce qui peut entraîner des modifications importantes. Par défaut, les versions personnalisées AMIs basées sur Amazon Linux sont automatiquement mises à jour vers la nouvelle version lors de leur publication. La méthode recommandée consiste à verrouiller votre AMI personnalisée sur une version Amazon Linux spécifique, ce qui vous permet de reporter la mise à jour jusqu'à ce que vous ayez testé la nouvelle version. Pour plus d'informations, consultez Comment verrouiller mon AMI avec une version spécifique ?.

Rubriques

- Création d'une AMI personnalisée à l'aide d'Amazon EC2
- Création d'une AMI Linux personnalisée à partir d'une AWS OpsWorks instance Stacks
- Créer une AMI Windows personnalisée

Création d'une AMI personnalisée à l'aide d'Amazon EC2

Le moyen le plus simple de créer une AMI personnalisée, et la seule option pour Windows AMIs, consiste à effectuer l'intégralité de la tâche à l'aide de la EC2 console ou de l'API Amazon. Pour plus de détails sur les étapes suivantes, consultez la section Création des vôtres AMIs.

Pour créer une AMI personnalisée à l'aide de EC2 la console ou de l'API Amazon

1. Créez une instance en utilisant une version 64 bits de l'un des logiciels compatibles avec AWS OpsWorks Stacks AMIs.
2. Personnalisez l'instance de l'étape 1 en la configurant, en installant les packages, et ainsi de suite. N'oubliez pas que tout ce que vous installez figurera sur chaque instance basée sur l'AMI ; par conséquent, n'incluez pas les éléments qui doivent être spécifiques à une instance particulière.
3. Arrêtez l'instance et créez une AMI personnalisée.

Création d'une AMI Linux personnalisée à partir d'une AWS OpsWorks instance Stacks

Pour utiliser une instance AWS OpsWorks Stacks Linux personnalisée afin de créer une AMI, sachez que chaque EC2 instance Amazon créée par OpsWorks inclut une identité unique. Si vous créez une AMI personnalisée à partir d'une telle instance, elle inclut cette identité, et toutes les instances basées sur l'AMI ont la même identité. Pour garantir que les instances basées sur votre AMI personnalisée aient une identité unique, vous devez supprimer l'identité de l'instance personnalisée avant de créer l'AMI.

Pour créer une AMI personnalisée à partir d'une AWS OpsWorks instance Stacks

1. Créez une pile Linux et ajoutez une ou plusieurs couches pour définir la configuration de l'instance personnalisée. Vous pouvez utiliser des couches intégrées, personnalisés selon le cas, aussi bien que des couches entièrement personnalisées. Pour de plus amples informations, veuillez consulter Personnalisation des piles AWS OpsWorks.
2. Modifiez les couches et désactivez-les AutoHealing.
3. Ajoutez une instance avec votre distribution Linux préférée à la couche ou aux couches et démarrez-la. Nous vous recommandons d'utiliser une instance basée sur Amazon EBS. Ouvrez la page de détails de l'instance et enregistrez son EC2 identifiant Amazon pour plus tard.
4. Lorsque l'instance est en ligne, connectez-vous avec SSH et procédez à l'une des quatre étapes suivantes, selon le système d'exploitation de votre instance.
5. Pour une instance Amazon Linux dans une pile Chef 11 ou Chef 12, ou une instance Red Hat Enterprise Linux 7 dans une pile Chef 11, procédez comme suit :

   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**
   >
   > Pour les instances d'une pile Chef 12, ajoutez les deux dossiers suivants à la commande :
   >
   > - `/var/chef`

    - /opt/chef

d. `sudo rpm -e opsworks-agent-ruby`

e. `sudo rpm -e chef`

6. Pour une instance Ubuntu 16.04 LTS ou 18.04 LTS d'une pile Chef 12, procédez comme suit.

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. Pour les autres versions Ubuntu prises en charge d'une pile Chef 12, procédez comme suit :

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. Pour une instance Red Hat Enterprise Linux 7 d'une pile Chef 12, procédez comme suit :

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. Cette étape dépend du type d'instance :

   - Pour une instance basée sur Amazon EBS, utilisez la console AWS OpsWorks Stacks pour arrêter l'instance et créer l'AMI comme décrit dans Création d'une AMI Linux basée sur Amazon EBS.
   - Pour une instance basée sur le stockage d'instance, créez l'AMI comme décrit dans Création d'une AMI Linux basée sur le stockage d'instance, puis utilisez AWS OpsWorks la console Stacks pour arrêter l'instance.

     Lorsque vous créez l'AMI, veillez à inclure les fichiers de certificat. Par exemple, vous pouvez appeler la commande `ec2-bundle-vol` avec l'argument `-i` défini sur `-i $(find /etc /usr /opt -name '*.pem' -o -name '*.crt' -o -name '*.gpg' | tr '\n' ',')`. Ne supprimez pas les clés publiques lors du regroupement. La commande par défaut `ec2-bundle-vol` gère la tâche.

10. Nettoyez votre pile en retournant à la console AWS OpsWorks Stacks et en supprimant l'instance de la pile.

Créer une AMI Windows personnalisée

Les procédures suivantes permettent de créer une AMIs version personnalisée pour Windows Server 2022 Base. Vous pouvez choisir d'autres systèmes d'exploitation Windows Server dans la console EC2 de gestion Amazon.

> **Important**
>
> À l'heure actuelle, l'agent AWS OpsWorks Stacks ne peut pas être installé sur des instances Windows qui utilisent une langue d'interface utilisateur système autre que l'anglais - États-Unis d'Amérique (en-US), et AWS OpsWorks Stacks ne peut pas les gérer.

Rubriques

- Création d'une AMI Windows personnalisée avec Sysprep
- Création d'une AMI Windows personnalisée sans Sysprep
- Ajout d'une nouvelle instance à l'aide d'une AMI Windows personnalisée

Création d'une AMI Windows personnalisée avec **`Sysprep`**

La création de Windows personnalisés à AMIs l'aide de Sysprep entraîne généralement un lancement d'instance plus lent, mais un processus plus propre. Le premier démarrage d'une instance créée à partir d'une image créée avec `Sysprep` prend plus de temps en raison des `Sysprep` activités, des redémarrages, du provisionnement des AWS OpsWorks Stacks et de la première exécution de AWS OpsWorks Stacks, y compris l'installation et la configuration. Suivez les étapes de création d'une AMI Windows personnalisée dans la EC2 console Amazon.

Pour créer une AMI Windows personnalisée avec Sysprep

1. Dans la EC2 console Amazon, choisissez Launch Instance.
2. Recherchez Microsoft Windows Server 2022 Base, puis choisissez Sélectionner.
3. Sélectionnez le type d'instance que vous voulez, puis choisissez Configure Instance Details (Configurer les détails de l'instance). Apportez les modifications de configuration à l'AMI, y compris les paramètres de nom de machine, de stockage et de groupes de sécurité. Choisissez Lancer.
4. Une fois que le processus de démarrage de l'instance a pris fin, obtenez votre mot de passe et connectez-vous à l'instance dans une fenêtre Windows Connexion Bureau à distance.
5. Sur l'écran d'accueil de Windows, choisissez Démarrer, puis commencez à taper **`ec2configservice`** jusqu'à ce que les résultats s'affichent sur la EC2ConfigServiceSettingsconsole. Ouvrez la console.
6. Dans l'onglet Général, assurez-vous que la case Activer UserData l'exécution est cochée (bien que cette option ne soit pas obligatoire`Sysprep`, elle est nécessaire pour que AWS OpsWorks Stacks installe son agent). Désactivez la case à cocher de l'option Set the computer name of the instance... (Définir le nom d'ordinateur de l'instance…), parce que cette option peut entraîner une boucle de redémarrage avec AWS OpsWorks Stacks.
7. Dans l'onglet Image, définissez le mot de passe administrateur sur Random pour permettre EC2 à Amazon de générer automatiquement un mot de passe que vous pouvez récupérer à l'aide d'une clé SSH, ou sur Specify pour spécifier votre propre mot de passe. `Sysprep`enregistre ce paramètre. Si vous spécifiez votre propre mot de passe, stockez le mot de passe dans un endroit pratique. Nous vous recommandons de ne pas choisir Keep Existing (Conserver l'existant).
8. Choisissez Apply (Appliquer), puis Shutdown with Sysprep (Arrêter avec Sysprep). Lorsque vous êtes invité à confirmer, choisissez Yes (Oui).
9. Une fois l'instance arrêtée, dans la EC2 console Amazon, cliquez avec le bouton droit sur l'instance dans la liste Instances, choisissez Image, puis Create Image.

10. Sur la page Create Image (Créer une image), fournissez le nom et la description de l'image, et spécifiez la configuration du volume. Lorsque vous avez terminé, choisissez Create Image (Créer une image).
11. Ouvrez la page Images et attendez que votre image passe de l'état pending (en attente) à l'état available (disponible). Votre nouvelle AMI est prêt à être utilisée.

Création d'une AMI Windows personnalisée sans **Sysprep**

Suivez les étapes de création d'une AMI Windows personnalisée dans la EC2 console Amazon.

Pour créer une AMI Windows personnalisée sans Sysprep

1. Dans la EC2 console Amazon, choisissez Launch Instance.
2. Recherchez Microsoft Windows Server 2022 Base, puis choisissez Sélectionner.
3. Sélectionnez le type d'instance que vous voulez, puis choisissez Configure Instance Details (Configurer les détails de l'instance). Apportez les modifications de configuration à l'AMI, y compris les paramètres de nom de machine, de stockage et de groupes de sécurité. Choisissez Lancer.
4. Une fois que le processus de démarrage de l'instance a pris fin, obtenez votre mot de passe et connectez-vous à l'instance dans une fenêtre Windows Connexion Bureau à distance.
5. Sur l'instance, ouvrez `C:\Program Files\Amazon\Ec2ConfigService\Settings\config.xml`, modifiez les deux paramètres suivants, puis enregistrez le fichier et fermez-le :
   - `Ec2SetPassword` sur `Enabled`
   - `Ec2HandleUserData` sur `Enabled`
6. Déconnectez-vous de la session Remote Desktop et revenez à la EC2 console Amazon.
7. Dans la liste Instances, arrêtez l'instance.
8. Une fois l'instance arrêtée, dans la EC2 console Amazon, cliquez avec le bouton droit sur l'instance dans la liste Instances, choisissez Image, puis Create Image.
9. Sur la page Create Image (Créer une image), fournissez le nom et la description de l'image, et spécifiez la configuration du volume. Lorsque vous avez terminé, choisissez Create Image (Créer une image).
10. Ouvrez la page Images et attendez que votre image passe de l'état pending (en attente) à l'état available (disponible). Votre nouvelle AMI est prêt à être utilisée.

Ajout d'une nouvelle instance à l'aide d'une AMI Windows personnalisée

Une fois que votre image est passée à l'état available (disponible), vous pouvez créer de nouvelles instances basées sur votre AMI Windows personnalisée. Lorsque vous choisissez Utiliser une AMI Windows personnalisée dans la liste des systèmes d'exploitation, AWS OpsWorks Stacks affiche une liste d'AMI personnalisées AMIs.

Pour ajouter une nouvelle instance basée sur une AMI Windows personnalisée

1. Lorsque votre nouvelle AMI est disponible, accédez à la console AWS OpsWorks Stacks, ouvrez la page Instances pour une pile Windows et choisissez + Instance en bas de la page pour ajouter une nouvelle instance.
2. Sous l'onglet New (Nouveau), choisissez Advanced (Avancé).
3. Dans la liste déroulante Operating system (Système d'exploitation), choisissez Use custom Windows AMI (Utiliser une AMI Windows personnalisée).
4. Dans la liste déroulante Custom AMI (AMI personnalisée), sélectionnez l'AMI que vous avez créée, puis choisissez Add Instance (Ajouter une instance).

Vous pouvez désormais démarrer et exécuter l'instance.

Démarrage, arrêt et redémarrage manuels des instances 24/7

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Note

Vous pouvez utiliser les instances 24/7 avec les piles Linux et Windows.

Après avoir ajouté une instance 24 h/24 et 7 j/7 à une couche, vous devez démarrer manuellement l'instance pour lancer l'instance Amazon Elastic Compute Cloud (Amazon EC2) correspondante et

l'arrêter manuellement pour mettre fin à l' EC2instance Amazon. Vous pouvez également redémarrer manuellement les instances qui ne fonctionnent pas correctement. AWS OpsWorks Stacks démarre et arrête automatiquement les instances basées sur le temps et sur la charge. Pour de plus amples informations, veuillez consulter Gestion de la charge avec des instances basées sur le temps et sur la charge.

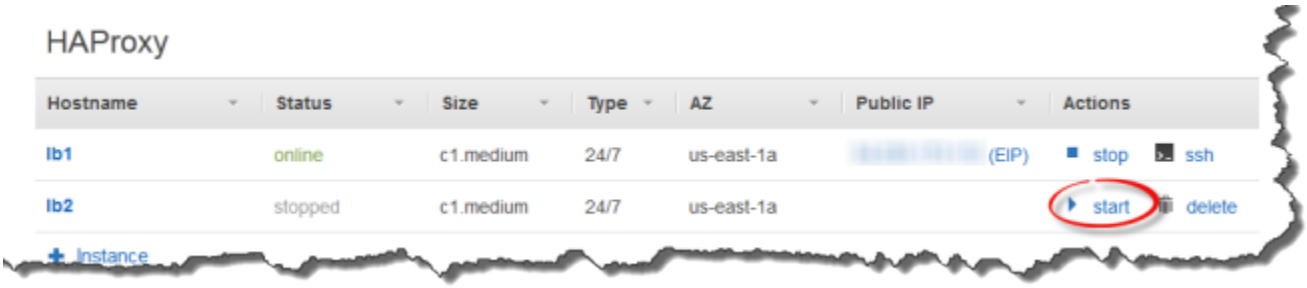
> **Important**
>
> AWS OpsWorks Les instances Stacks doivent être démarrées, arrêtées et redémarrées uniquement dans la AWS OpsWorks console. AWS OpsWorks ne reconnaît pas les opérations de démarrage, d'arrêt ou de redémarrage effectuées dans la EC2 console Amazon.

Rubriques

- Démarrage ou redémarrage d'une instance
- Arrêt d'une instance
- Redémarrage d'une instance

Démarrage ou redémarrage d'une instance

Pour démarrer une nouvelle instance, sur la page Instances, cliquez sur start (démarrer) dans la colonne Actions de l'instance.



Vous pouvez aussi créer plusieurs instances et toutes les redémarrer simultanément en cliquant sur Start all Instances (Démarrer toutes les instances).

Après avoir démarré l'instance, AWS OpsWorks Stacks lance une EC2 instance Amazon et démarre le système d'exploitation. Généralement, le processus de démarrage ne prend que quelques minutes et se montre un peu plus lent pour les instances Windows que pour les instances Linux. Au fur et à

mesure que le démarrage progresse, le champ Status (Statut) de l'instance affiche la série de valeurs suivantes :

1. demandé - AWS OpsWorks Stacks a appelé le EC2 service Amazon pour créer l' EC2 instance Amazon.
2. en attente : AWS OpsWorks Stacks attend le démarrage de l' EC2instance Amazon.
3. démarrage : l' EC2 instance Amazon est en train de démarrer.
4. running_setup - AWS OpsWorks Stacks a déclenché l'événement de configuration et exécute le Setup de la couche, suivies de ses recettes Deploy les recettes. Pour de plus amples informations, veuillez consulter Exécution des recettes. Si vous avez ajouté des livres de recettes personnalisés à la pile, AWS OpsWorks Stacks installe la version actuelle depuis votre dépôt avant d'exécuter le Setup and Deploy les recettes.
5. online (en ligne) - l'instance est prête à être utilisée.

Lorsque Status (Statut) devient online (en ligne), l'instance est pleinement opérationnelle.

- Si un équilibreur de charge est attaché à la couche, AWS OpsWorks Stacks y ajoute l'instance.
- AWS OpsWorks Stacks déclenche un Configure qui exécute les recettes Configure les recettes.

  Si nécessaire, ces recettes mettent à jour l'instance pour accueillir la nouvelle instance.
- AWS OpsWorks Stacks remplace l'action de démarrage de l'instance par l'action stop, que vous pouvez utiliser pour arrêter l'instance.

Si l'instance n'a pas démarré correctement ou que les recettes Setup ont échoué, le statut est défini sur start_failed ou setup_failed, respectivement. Vous pouvez examiner les journaux afin de déterminer la cause. Pour de plus amples informations, veuillez consulter Guide de débogage et dépannage.

Une instance arrêtée demeure dans le cadre de la pile et conserve toutes les ressources. Par exemple, les volumes Amazon EBS et les adresses IP Elastic sont toujours associés à une instance arrêtée. Vous pouvez redémarrer une instance arrêtée en choisissant start (démarrer) dans la colonne Actions de l'instance. Le redémarrage d'une instance arrêtée effectue les opérations suivantes :

- Instances basées sur le stockage d'instances : AWS OpsWorks Stacks lance une nouvelle EC2 instance Amazon avec la même configuration.

- Instances soutenues par Amazon EBS : AWS OpsWorks Stacks redémarre l'instance EC2 Amazon, qui réattache le volume racine.

Une fois le démarrage de l'instance terminé, AWS OpsWorks Stacks installe les mises à jour du système d'exploitation et exécute le Setup and Deploy recettes, comme lors du démarrage initial. AWS OpsWorks Stacks effectue également les opérations suivantes pour les instances redémarrées, le cas échéant.

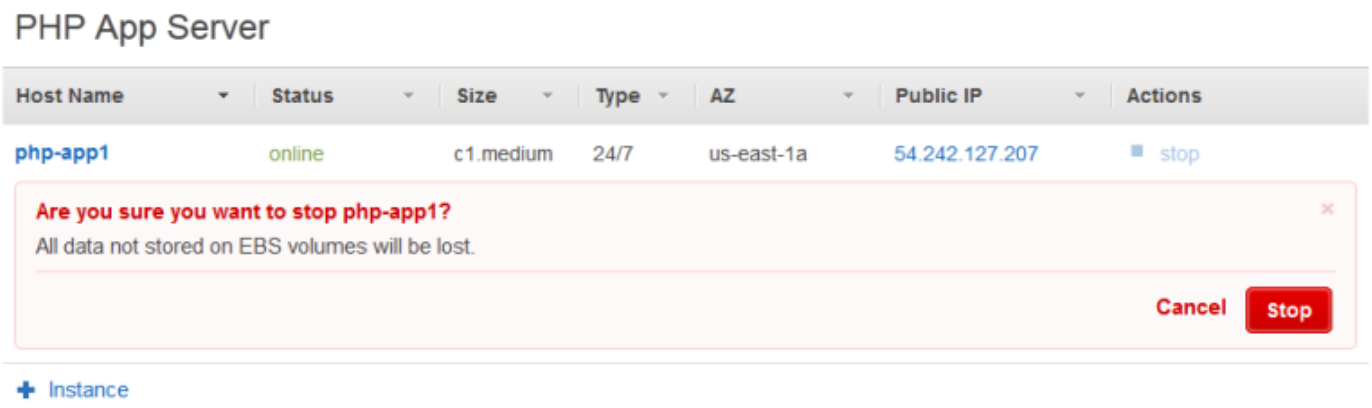
- Réassocie les adresses IP Elastic.
- Rattache les volumes Amazon Elastic Block Store (Amazon EBS).
- Pour les instances basées sur le stockage d'instance, installe les dernières versions des livres de recettes.

  Les instances basées sur Amazon EBS continuent d'utiliser les livres de recettes personnalisés stockés sur le volume racine. Si vos livres de recettes personnalisés ont été modifiés depuis que vous avez arrêté l'instance, vous devez les mettre à jour manuellement une fois que l'instance est en ligne. Pour de plus amples informations, veuillez consulter Mise à jour des livres de recettes personnalisés.

> **Note**
>
> Plusieurs minutes peuvent être nécessaires pour qu'une adresse IP Elastic soit réassociée à une instance redémarrée. Sachez que le paramètre Elastic IP de l'instance représente les métadonnées et indique simplement que l'adresse doit être associée à l'instance. Le paramètre Public IP (Adresse IP publique) reflète l'état de l'instance et peut être vide au départ. Lorsque l'adresse IP Elastic est associée à l'instance, l'adresse est attribuée au paramètre Public IP (Adresse IP publique), suivi par (EIP).

Arrêt d'une instance

Sur la page Instances, cliquez sur Arrêter dans la colonne Actions de l'instance, qui indique à AWS OpsWorks Stacks d'exécuter les recettes d'arrêt et de mettre fin à l' EC2 instance.



Vous pouvez aussi fermer chaque instance de la pile en cliquant sur Arrêter toutes les instances.

Après avoir arrêté l'instance, AWS OpsWorks Stacks exécute plusieurs tâches :

1. Si un équilibreur de charge Elastic Load Balancing est attaché à la couche de l'instance, AWS OpsWorks Stacks annule l'enregistrement de l'instance.

   Si la couche prend en charge la fonction de drainage de la connexion de l'équilibreur de charge, AWS OpsWorks Stacks retarde le déclenchement de l'événement Shutdown événement jusqu'à ce que le vidage de la connexion soit terminé. Pour de plus amples informations, veuillez consulter Couche Elastic Load Balancing.
2. AWS OpsWorks Stacks déclenche un Shutdown qui exécute les recettes Shutdown les recettes.
3. Après avoir déclenché l'événement Shutdown événement, AWS OpsWorks Stacks attend un temps spécifié pour autoriser Shutdown aient le temps de se terminer, puis effectue les opérations suivantes :
   - Met fin aux instances basées sur le stockage d'instance, ce qui supprime toutes les données.
   - Arrête les instances soutenues par Amazon EBS, ce qui préserve les données sur le volume racine.

   Pour plus d'informations sur le stockage d'instance, consultez Stockage.

> **Note**
>
> Le paramètre de délai d'attente de fermeture par défaut est de 120 secondes. Si vos recettes Shutdown ont besoin de plus de temps, vous pouvez modifier la configuration de la couche pour changer le paramètre.

Vous pouvez surveiller le processus d'arrêt en examinant la colonne Status (Statut) de l'instance. Au fur et à mesure que l'arrêt progresse, les valeurs suivantes s'affichent :

1. terminaison - AWS OpsWorks Stacks met fin à l'instance Amazon. EC2
2. shutting_down - AWS OpsWorks Stacks exécute le Shutdown les recettes.
3. terminée : l' EC2 instance Amazon est résiliée.
4. stopped (arrêté) - l'instance est arrêtée.

Redémarrage d'une instance

Sur la page Instances, cliquez sur le nom de l'instance défectueuse pour ouvrir la page des détails, puis cliquez sur Reboot (Redémarrer).



Cette commande effectue un redémarrage progressif de l' EC2 instance Amazon associée. Elle ne supprime pas les données de l'instance, même dans le cas d'instances basées sur le stockage d'instance, et ne déclenche aucun événement de cycle de vie.

> **Note**
>
> Pour que les AWS OpsWorks Stacks remplacent automatiquement les instances défaillantes, activez la réparation automatique. Pour de plus amples informations, veuillez consulter Utilisation de la réparation automatique.

Gestion de la charge avec des instances basées sur le temps et sur la charge

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Comme le trafic entrant varie, votre pile peut-être avoir trop peu d'instances pour gérer confortablement la charge ou plus d'instances que nécessaire. Vous pouvez gagner du temps et de l'argent en utilisant les instances à date définie ou à charge définie, et augmenter ou réduire automatiquement les instances d'une couche de telle sorte que vous ayez toujours assez d'instances pour gérer de façon adéquate le trafic entrant sans avoir à payer la capacité superflue. Il n'y a pas besoin de surveiller les charges du serveur, ou de démarrer ou arrêter manuellement les instances. En outre, les instances à date définie et à charge définie répartissent, mettent à l'échelle et équilibrent automatiquement les applications sur plusieurs zones de disponibilité au sein d'une région, ce qui vous procure évolutivité et redondance géographique.

Le dimensionnement automatique repose sur deux types d'instances qui ajustent les instances en ligne d'une couche selon différents critères :

- Instances à date définie

  Elles permettent à une pile de gérer les charges qui suivent un modèle prévisible en incluant les instances qui ne s'exécutent qu'à certaines heures ou que certains jours. Par exemple, vous pouvez démarrer certaines instances après 18 h 00 pour exécuter les tâches nocturnes de sauvegarde ou arrêter certaines instances le week-end lorsque le trafic est plus lent.
- Instances à charge définie

  Elles permettent à une pile de gérer les charges variables en démarrant des instances supplémentaires lorsque le trafic est élevé et en arrêtant les instances lorsque le trafic est faible, en fonction d'une ou de plusieurs métriques de charge. Par exemple, vous pouvez demander à AWS OpsWorks Stacks de démarrer des instances lorsque l'utilisation moyenne du processeur dépasse 80 % et d'arrêter des instances lorsque la charge moyenne du processeur tombe en dessous de 60 %.

Aussi bien les instances à date définie que les instances à charge définie sont prises en charge avec les piles Linux, alors que seules les instances à date définie sont prises en charge avec les piles Windows.

Contrairement aux instances 24/7, que vous devez arrêter et démarrer manuellement, vous ne démarrez ni n'arrêtez vous-même les instances à date définie ou à charge définie. Au lieu de cela,

vous configurez les instances et AWS OpsWorks Stacks les démarre ou les arrête en fonction de leur configuration. Par exemple, vous configurez des instances basées sur le temps pour qu'elles démarrent et s'arrêtent selon un calendrier spécifié. AWS OpsWorks Stacks démarre et arrête ensuite les instances conformément à cette configuration.

Une pratique courante consiste à utiliser ensemble les trois types d'instance, comme ci-après.

- Un ensemble d'instances 24/7 pour gérer la charge de base. Généralement, vous démarrez simplement ces instances et les laissez s'exécuter en continu.
- Un ensemble d'instances basées sur le temps, que AWS OpsWorks Stacks démarre et arrête pour gérer les variations prévisibles du trafic. Par exemple, si votre trafic est le plus élevé pendant les heures de travail, vous configurez les instances à date définie pour qu'elles commencent le matin et les arrêtez dans la soirée.
- Un ensemble d'instances basées sur la charge, que AWS OpsWorks Stacks démarre et arrête pour gérer les variations imprévisibles du trafic. AWS OpsWorks Stacks les démarre lorsque la charge approche la capacité des instances basées sur le temps et 24 heures sur 24, 7 jours sur 7, et les arrête lorsque le trafic revient à la normale.

Pour plus d'informations sur l'utilisation des heures de dimensionnement, consultez Optimisation du nombre de serveurs.

> **Note**
>
> Si vous avez créé des applications pour la couche des instances ou créé des livres de recettes personnalisés, AWS OpsWorks Stacks déploie automatiquement la dernière version sur les instances basées sur le temps et sur la charge lors de leur premier démarrage. Cependant, AWS OpsWorks Stacks ne déploie pas nécessairement les derniers livres de recettes sur les instances hors ligne redémarrées. Pour plus d'informations, consultez Modification des applications et Mise à jour des livres de recettes personnalisés.

Rubriques

- Utilisation du dimensionnement automatique basé sur le temps
- Utilisation du dimensionnement automatique basé sur la charge
- En quoi la mise à l'échelle basée sur la charge diffère de la réparation automatique

Utilisation du dimensionnement automatique basé sur le temps

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Le dimensionnement basé sur le temps vous permet de contrôler le nombre d'instances qu'une couche doit avoir en ligne à certains moments de la journée ou certains jours de la semaine en démarrant ou en arrêtant des instances selon un calendrier défini. AWS OpsWorks Stacks vérifie toutes les deux minutes et démarre ou arrête les instances selon les besoins. Vous spécifiez la planification séparément pour chaque instance, comme suit :

- Heure du jour. Vous pouvez avoir plus d'instances qui s'exécutent le jour que la nuit, par exemple.
- Jour de la semaine. Vous pouvez avoir plus d'instances qui s'exécutent les jours de la semaine que le week-end, par exemple.

> **Note**
>
> Vous ne pouvez pas spécifier de dates particulières.

Rubriques

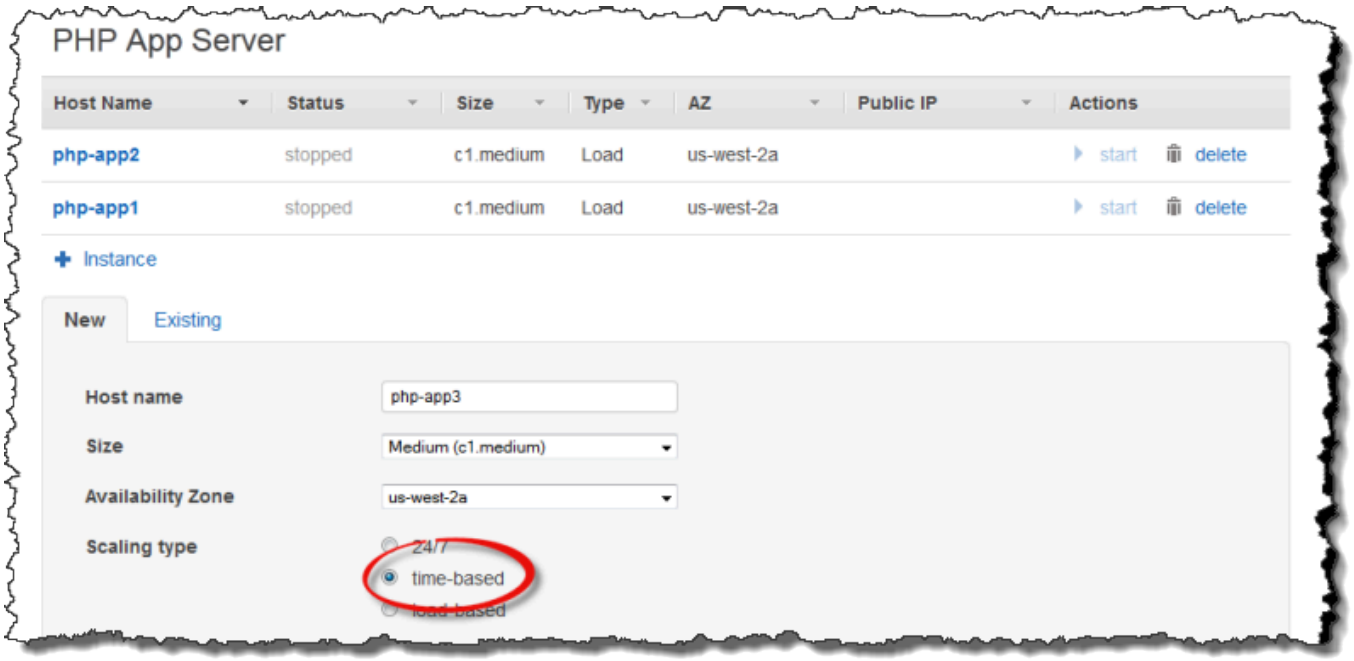
- Ajouter une instance basée sur le temps à une couche
- Configuration d'une instance basée sur le temps

Ajouter une instance basée sur le temps à une couche

Vous pouvez ajouter une nouvelle instance à date définie à une couche, ou utiliser une instance existante.

Pour ajouter une nouvelle instance à date définie

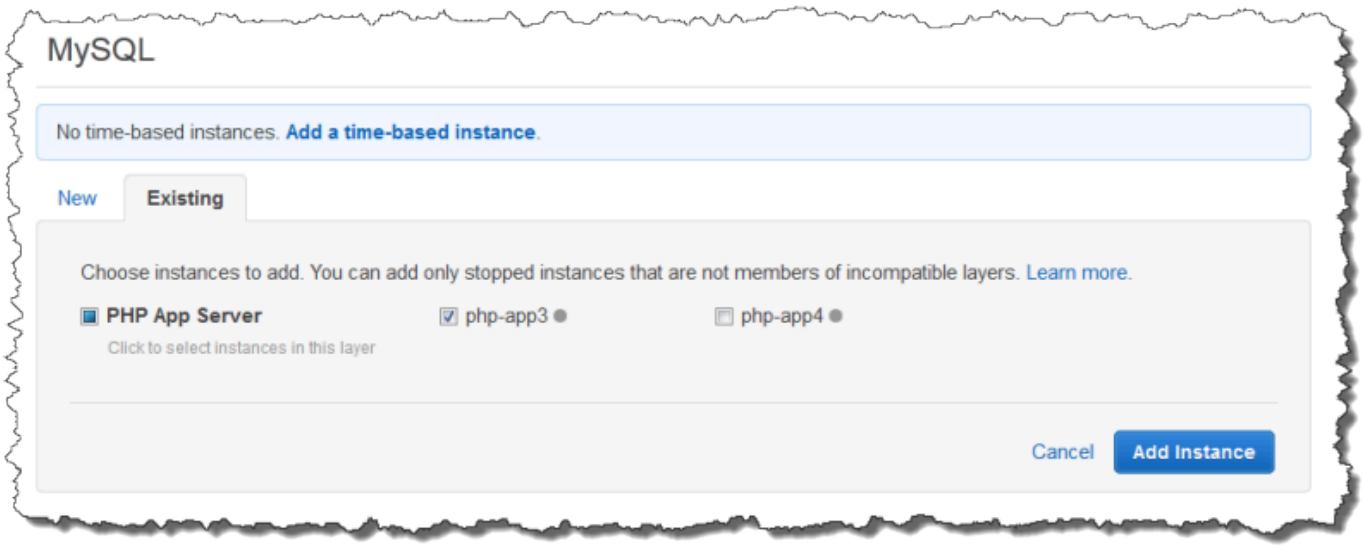
1. Sur la page Instances, choisissez + Instance pour ajouter une instance. Dans l'onglet Nouveau, choisissez Avancé, puis sélectionnez Basé sur le temps.



2. Configurez l'instance. Choisissez ensuite Add Instance pour ajouter l'instance à la couche.

Pour ajouter une instance existante à date définie à une couche

1. Sur la page Instances temporelles, choisissez + Instance si une couche possède déjà une instance temporelle. Sinon, choisissez Ajouter une instance basée sur le temps. Choisissez ensuite l'onglet Existant.



2. Dans l'onglet Existing, choisissez une instance dans la liste. La liste ne montre que les instances à date définie.

> **Note**
>
> Si vous changez d'avis quant à l'utilisation d'une instance existante, dans l'onglet Nouveau, créez une nouvelle instance, comme décrit dans la procédure précédente.

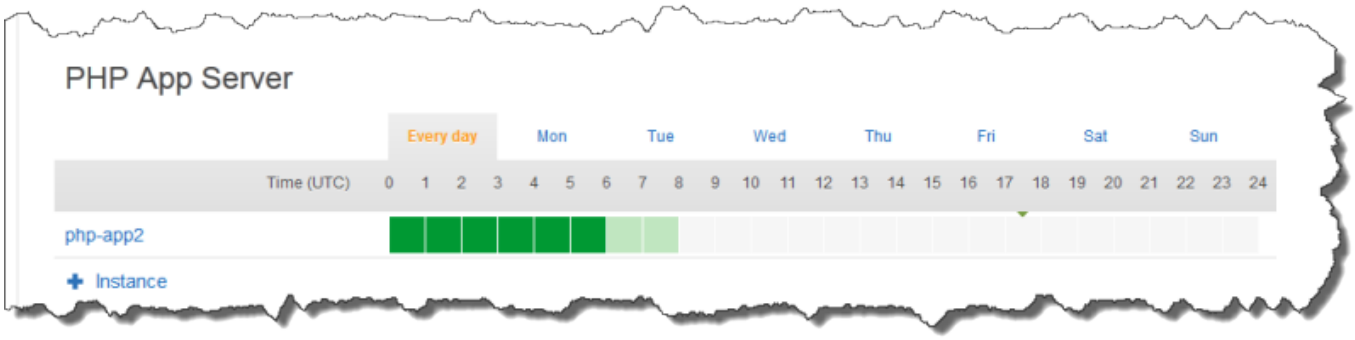
3. Choisissez Ajouter une instance pour ajouter l'instance à la couche.

Configuration d'une instance basée sur le temps

Une fois que vous avez ajouté une instance à date définie à une couche, vous configurez son calendrier comme suit.

Pour configurer une instance à date définie

1. Dans le volet de navigation, sous Instances, choisissez Time-based.
2. Spécifiez les périodes en ligne pour chaque instance basée sur le temps en remplissant les cases appropriées en dessous de l'heure souhaitée.
   - Pour utiliser le même planning tous les jours, cliquez sur l'onglet Tous les jours, puis spécifiez les périodes en ligne.
   - Pour utiliser des horaires différents selon les jours, choisissez chaque jour, puis les périodes appropriées.



> **Note**
>
> Veillez à tenir compte du temps nécessaire au démarrage d'une instance et à ce que AWS OpsWorks Stacks vérifie uniquement toutes les quelques minutes si les instances doivent être démarrées ou arrêtées. Par exemple, si une instance doit s'exécuter à 01:00 UTC, démarrez-la à 0:00 UTC. Dans le cas contraire, AWS OpsWorks Stacks risque de ne démarrer l'instance que quelques minutes après 1h00 UTC, et l'instance mettra encore plusieurs minutes à être en ligne.

Vous pouvez modifier les périodes de connexion d'une instance à tout moment en suivant les étapes précédentes. La prochaine fois que AWS OpsWorks Stacks vérifie, il utilise le nouveau calendrier pour déterminer s'il faut démarrer ou arrêter les instances.

> **Note**
>
> Vous pouvez ajouter une nouvelle instance temporelle à une couche en ouvrant la page basée sur le temps et en choisissant Ajouter une instance temporelle (si vous n'avez pas encore ajouté d'instance temporelle à la couche) ou + Instance (si la couche possède déjà une ou plusieurs instances temporelles). Configurez ensuite l'instance comme décrit dans les procédures précédentes.

Utilisation du dimensionnement automatique basé sur la charge

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les instances basées sur la charge vous permettent de démarrer ou d'arrêter rapidement des instances en réponse à l'évolution du trafic entrant. AWS OpsWorks Stacks utilise CloudWatch les données Amazon pour calculer les métriques suivantes pour chaque couche, qui représentent les valeurs moyennes de toutes les instances de la couche :

- CPU : consommation UC moyenne, telle que 80 %
- Memory : consommation mémoire moyenne, telle que 60 %
- Charge : opérations de calcul qu'un système effectue en moyenne en une minute.

Vous définissez les seuils upscaling (agrandissement) et downscaling (réduction) pour tout ou partie des métriques. Vous pouvez également utiliser des CloudWatch alarmes personnalisées comme seuils.

Le franchissement d'un seuil déclenche un événement de dimensionnement. Vous déterminez comment AWS OpsWorks Stacks répond aux événements de dimensionnement en spécifiant les éléments suivants :

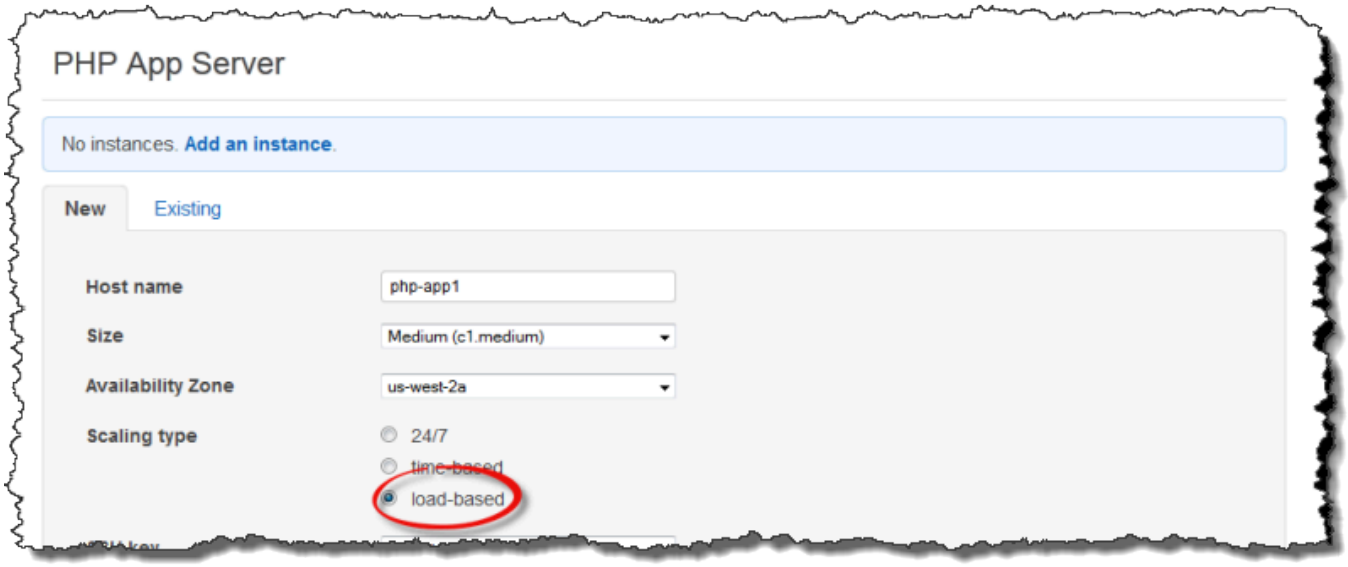
- Nombre d'instances à démarrer ou arrêter.
- Combien de temps les AWS OpsWorks Stacks doivent attendre après avoir dépassé un seuil avant de démarrer ou de supprimer des instances. Par exemple, l'utilisation de l'UC doit dépasser le seuil pendant au moins 15 minutes. Cette valeur vous permet d'ignorer les fluctuations de trafic brèves.
- Combien de temps les AWS OpsWorks Stacks doivent attendre après le démarrage ou l'arrêt des instances avant de surveiller à nouveau les métriques. Généralement, vous voulez autoriser assez de temps pour que les instances démarrées soient en ligne ou que les instances interrompues s'arrêtent avant de déterminer si la couche continue de dépasser un seuil.

Lorsqu'un événement de dimensionnement se produit, AWS OpsWorks Stacks démarre ou arrête uniquement les instances basées sur la charge. Il ne démarre ni n'arrête les instances 24/7 ou les instances à date définie.

> **Note**
>
> Le dimensionnement automatique à charge définie ne crée pas d'instances ; il démarre et arrête uniquement les instances que vous avez créées. Par conséquent, vous devez provisionner à l'avance un nombre suffisant d'instances à charge définie pour gérer la charge maximale prévue.

Pour créer une instance à charge définie

1. Sur la page Instances, choisissez +Instance pour ajouter une instance. Choisissez Avancé, puis sélectionnez Basé sur la charge.



2. Configurez l'instance, puis choisissez Ajouter une instance pour ajouter l'instance à la couche.

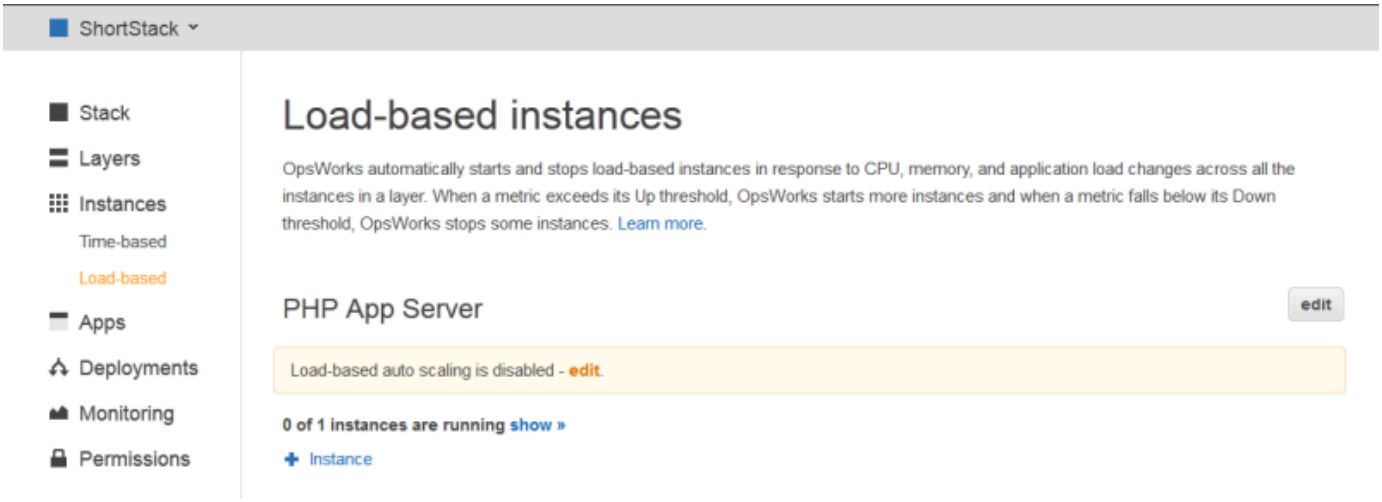
Répétez cette procédure jusqu'à ce que vous ayez créé un nombre suffisant d'instances. Vous pourrez ajouter ou supprimer des instances plus tard, comme requis.

Une fois que vous avez ajouté des instances à charge définie à une couche, vous devez activer le dimensionnement à charge définie et spécifier la configuration. La configuration du dimensionnement à date définie est une propriété de la couche, pas une propriété de l'instance, qui spécifie quand une

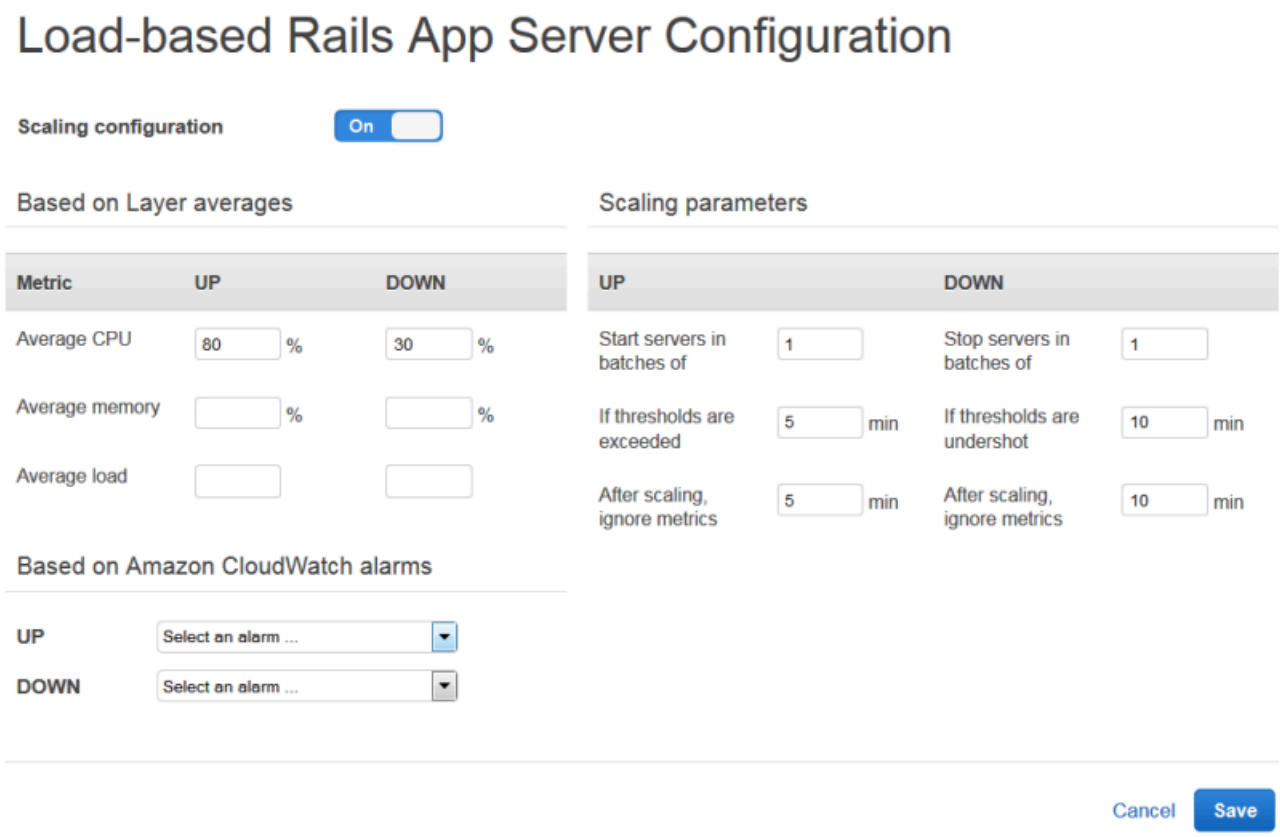
couche doit démarrer ou arrêter ses instances à charge définie. Cette propriété doit être spécifiée séparément pour chaque couche qui utilise les instances à charge définie.

Pour activer et configurer le dimensionnement automatique à charge définie

1. Dans le volet de navigation, sous Instances, sélectionnez Basé sur la charge, puis sélectionnez Modifier pour la couche appropriée.



2. Activez la mise à l'échelle automatique basée sur la charge sur Activé. Puis, définissez les paramètres de seuil et de dimensionnement de façon à définir comment et quand ajouter ou supprimer des instances.



Seuils moyens par couche

Vous pouvez dimensionner les seuils en fonction des valeurs suivantes, dont la moyenne est calculée sur l'ensemble des instances de la couche.

- CPU moyen : utilisation moyenne du processeur de la couche, en pourcentage du total.
- Mémoire moyenne : utilisation moyenne de la mémoire de la couche, en pourcentage du total.
- Charge moyenne : charge moyenne de la couche.

  Pour plus d'informations sur le mode de calcul de la charge, consultez Load (computing) sur Wikipedia.

Le franchissement d'un seuil entraîne un événement de dimensionnement, une mise à l'échelle supérieure si davantage d'instances sont nécessaires et une réduction de la taille si moins d'instances sont nécessaires. AWS OpsWorks Stacks ajoute ou supprime ensuite des instances en fonction des paramètres de mise à l'échelle.

CloudWatch Alarmes personnalisées

Vous pouvez utiliser jusqu'à cinq CloudWatch alarmes personnalisées comme seuils d'augmentation ou de réduction d'échelle. Ils doivent être dans la même région que la pile. Pour plus d'informations sur la création d'alarmes personnalisées, consultez Creating Amazon CloudWatch Alarms.

> **Note**
>
> Pour utiliser les alarmes personnalisées, vous devez mettre à jour votre rôle de service pour permettre `cloudwatch:DescribeAlarms`. Vous pouvez soit demander à AWS OpsWorks Stacks de mettre à jour le rôle pour vous la première fois que vous utilisez cette fonctionnalité, soit le modifier manuellement. Pour de plus amples informations, veuillez consulter Permettre à AWS OpsWorks Stacks d'agir en votre nom.
>
> Lorsque plusieurs alarmes sont configurées pour une configuration basée sur la charge, si une alarme est dans l'état d'alarme `INSUFFICIENT_DATA` métrique, le dimensionnement de l'instance basé sur la charge ne peut pas se produire même si une autre alarme est dans cet état. `ALARM` Le dimensionnement automatique ne peut se poursuivre que si toutes les alarmes sont à l'`ALARM`état `OK` ou. Pour plus d'informations sur l'utilisation des CloudWatch alarmes Amazon, consultez la section Utilisation des CloudWatch alarmes Amazon dans le guide de CloudWatch l'utilisateur Amazon.

Paramètres de dimensionnement

Les paramètres suivants contrôlent la façon dont AWS OpsWorks Stacks gère les événements de dimensionnement.

- Démarrer les serveurs par lots de : nombre d'instances à ajouter ou à supprimer lorsque l'événement de dimensionnement se produit.
- Si les seuils sont dépassés : durée (en minutes) pendant laquelle la charge doit rester au-dessus d'un seuil d'augmentation ou en dessous d'un seuil de réduction d'échelle avant que AWS OpsWorks Stacks ne déclenche un événement de dimensionnement.
- Après le dimensionnement, ignorez les métriques : délai (en minutes) après la survenue d'un événement de dimensionnement pendant lequel AWS OpsWorks Stacks doit ignorer les métriques et supprimer les événements de dimensionnement supplémentaires.

Par exemple, AWS OpsWorks Stacks ajoute de nouvelles instances à la suite d'un événement de montée en charge, mais les instances ne commenceront pas à réduire la charge tant qu'elles n'auront pas été démarrées et configurées. Il est inutile de lever des événements de dimensionnement supplémentaires tant que les nouvelles instances ne sont pas en ligne et de gérer les demandes, ce qui prend généralement plusieurs minutes. Ce paramètre vous permet de demander à AWS OpsWorks Stacks de supprimer les événements de dimensionnement assez longtemps pour obtenir les nouvelles instances en ligne.

Vous pouvez augmenter ce paramètre pour éviter les fluctuations soudaines de mise à l'échelle lorsque les moyennes des couches, telles que le processeur moyen, la mémoire moyenne ou la charge moyenne, sont temporairement en désaccord.

Par exemple, si l'utilisation du processeur est supérieure à la limite et que l'utilisation de la mémoire est proche de la réduction d'échelle, un événement de montée en gamme d'instance peut être immédiatement suivi d'un événement de réduction de la mémoire. Pour éviter cela, vous pouvez augmenter le nombre de minutes dans le paramètre Après le dimensionnement, ignorer les métriques. Dans cet exemple, le dimensionnement du processeur se produirait, mais pas l'événement de réduction de la mémoire.

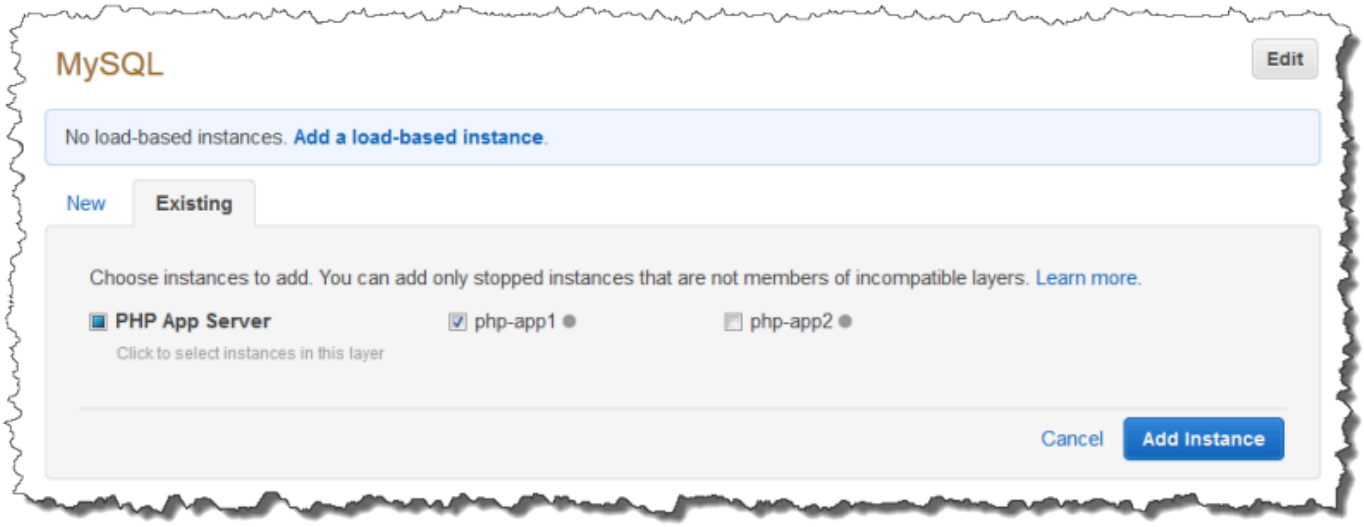
3. Pour ajouter des instances supplémentaires basées sur la charge, choisissez + Instance, configurez les paramètres, puis choisissez Ajouter une instance. Répétez jusqu'à ce que vous ayez assez d'instances à charge définie pour gérer votre charge maximale prévue. Ensuite, choisissez Save (Enregistrer).

> **Note**
>
> Vous pouvez également ajouter une nouvelle instance basée sur la charge à une couche en ouvrant la page basée sur la charge et en choisissant Ajouter une instance basée sur la charge (si vous n'avez pas encore ajouté d'instance basée sur la charge à la couche) ou + Instance (si la couche possède déjà une ou plusieurs instances basées sur la charge). Puis, configurez l'instance comme indiqué plus tôt dans cette section.

Pour ajouter une instance existante à charge définie à une couche

1. Dans le volet de navigation, sous Instances, choisissez Load-based.

2. Si vous avez déjà activé la mise à l'échelle automatique basée sur la charge pour une couche, choisissez + Instance. Sinon, choisissez Ajouter une instance basée sur la charge. Choisissez l'onglet Existant.



3. Dans l'onglet Existing, choisissez une instance. La liste affiche uniquement les instances à charge définie.

> **Note**
>
> Si vous changez d'avis quant à l'utilisation d'une instance existante, dans l'onglet Nouveau, créez une nouvelle instance comme décrit dans la procédure précédente.

4. Choisissez Ajouter une instance pour ajouter l'instance à la couche.

Vous pouvez modifier la configuration pour activer ou désactiver le dimensionnement automatique à charge définie à tout moment.

Pour désactiver le dimensionnement automatique à charge définie

1. Dans le volet de navigation, sous Instances, sélectionnez Basé sur la charge, puis sélectionnez Modifier pour la couche appropriée.
2. Basculez la mise à l'échelle automatique basée sur la charge activée sur Non.

En quoi la mise à l'échelle basée sur la charge diffère de la réparation automatique

Le dimensionnement automatique à charge définie utilise les métriques de charge dont la moyenne est calculée sur toutes les instances en cours d'exécution. Si les métriques restent entre les seuils spécifiés, AWS OpsWorks Stacks ne démarre ni n'arrête aucune instance. Avec la guérison automatique, en revanche, AWS OpsWorks Stacks démarre automatiquement une nouvelle instance avec la même configuration lorsqu'une instance cesse de répondre. L'instance peut ne pas être en mesure de répondre en raison d'un problème de réseau ou d'un problème lié à l'instance.

Supposons, par exemple, que le seuil de mise à l'échelle du processeur soit de 80 % et qu'une instance cesse de répondre.

- Si la réparation automatique est désactivée et que les instances en cours d'exécution restantes peuvent maintenir l'utilisation moyenne du processeur en dessous de 80 %, AWS OpsWorks Stacks ne démarre pas de nouvelle instance. Il démarre une instance de remplacement uniquement si l'utilisation moyenne de l'UC entre les instances restantes est supérieure à 80 %.
- Si la réparation automatique est activée, AWS OpsWorks Stacks démarre une instance de remplacement quels que soient les seuils de charge.

Utilisation des ressources de calcul créées en dehors d' AWS OpsWorks Stacks

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Note

Cette fonctionnalité est prise en charge uniquement pour les piles Linux.

instancesdécrit comment utiliser AWS OpsWorks Stacks pour créer et gérer des groupes d'instances Amazon Elastic Compute Cloud (Amazon EC2). Vous pouvez également intégrer des ressources informatiques Linux dans une pile créée en dehors de AWS OpsWorks Stacks :

- EC2 Instances Amazon que vous avez créées directement à l'aide de la EC2 console, de la CLI ou de l'API Amazon.
- Instances locales s'exécutant sur votre propre matériel, y compris les instances s'exécutant dans les machines virtuelles.

Ces ressources informatiques deviennent des instances AWS OpsWorks gérées par Stacks, et vous pouvez les gérer de la même manière que les instances Stacks classiques AWS OpsWorks :

- Gérer les autorisations des utilisateurs : vous pouvez utiliser la gestion des utilisateurs de AWS OpsWorks Stacks pour spécifier quels utilisateurs sont autorisés à accéder à vos piles, quelles actions ils sont autorisés à effectuer sur les instances de la pile et s'ils disposent d'un accès SSH et de privilèges sudo.
- Automatiser les tâches : vous pouvez demander à AWS OpsWorks Stacks d'exécuter des recettes Chef personnalisées pour effectuer des tâches telles que l'exécution de scripts sur une ou toutes les instances d'une pile à l'aide d'une seule commande.

  Si vous attribuez l'instance à une couche, AWS OpsWorks Stacks exécute automatiquement un ensemble spécifique de recettes Chef sur l'instance à des moments clés de son cycle de vie, y compris vos recettes personnalisées. Notez que vous ne pouvez attribuer des EC2 instances Amazon enregistrées qu'à des couches personnalisées.
- Gestion des ressources : une pile vous permet de regrouper et de gérer les ressources dans un Région AWS, et le OpsWorks tableau de bord indique l'état de vos piles dans toutes les régions.
- Installer des packages — Vous pouvez utiliser des recettes Chef pour installer des packages sur n'importe quelle instance d'une pile.
- Mettre à jour le système d'exploitation — AWS OpsWorks Stacks fournit un moyen simple d'installer les correctifs de sécurité et les mises à jour du système d'exploitation sur les instances d'une pile.
- Déployer des applications : AWS OpsWorks Stacks déploie les applications de manière cohérente sur toutes les instances de serveur d'applications de la pile.
- Surveillance — AWS OpsWorks Stacks crée des CloudWatchmétriques personnalisées pour surveiller toutes les instances de votre stack.

Pour plus d'informations sur les tarifs, consultez la section OpsWorks Tarification AWS.

Voici la procédure de base pour utiliser une instance enregistrée.

1. Enregistrez l'instance auprès d'une pile.

   L'instance fait désormais partie de la pile et est gérée par AWS OpsWorks Stacks.
2. Le cas échéant, assignez l'instance à une couche.

   Cette étape vous permet de tirer pleinement parti des fonctionnalités de gestion de AWS OpsWorks Stacks. Vous pouvez attribuer des instances sur site enregistrées à n'importe quelle couche ; les EC2 instances Amazon enregistrées ne peuvent être attribuées qu'à des couches personnalisées.
3. Utilisez AWS OpsWorks Stacks pour gérer l'instance.
4. Lorsque vous n'avez plus besoin de l'instance dans la pile, désenregistrez-la, ce qui la supprime de AWS OpsWorks Stacks.

Les sections suivantes décrivent le processus en détail.

Rubriques

- Enregistrement d'une instance auprès d'un AWS OpsWorks Stacks Stacks
- Gestion des instances enregistrées
- Attribution d'une instance enregistrée à une couche
- Annulation de l'enregistrement d'une instance
- Annulation de l'enregistrement d'une instance
- Cycle de vie des instances enregistrées

Enregistrement d'une instance auprès d'un AWS OpsWorks Stacks Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette fonctionnalité est prise en charge uniquement pour les piles Linux.

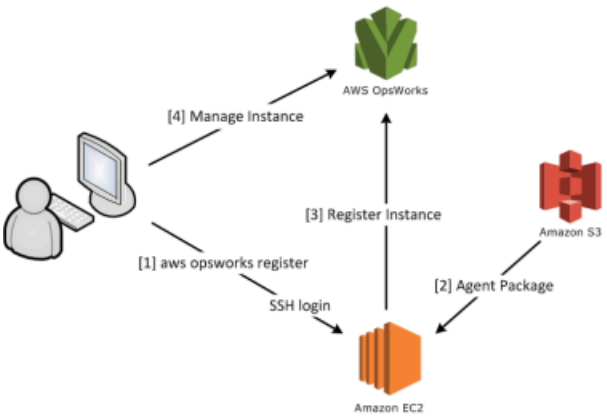
Pour enregistrer une instance en dehors de AWS OpsWorks Stacks, vous devez exécuter la AWS CLI aws opsworks register commande. Vous pouvez exécuter cette commande à partir de l'instance que vous souhaitez enregistrer, ou à partir d'un autre ordinateur. Vous appliquez les `AWSOpsWorksRegisterCLI_OnPremises` politiques `AWSOpsWorksRegisterCLI_EC2` ou à un utilisateur ou à un groupe pour accorder les autorisations requises pour l'enregistrement EC2 ou AWS CLI pour les instances sur site, respectivement. Ces politiques nécessitent la version 1.16.180 AWS CLI ou une version ultérieure.

> **Note**
>
> Pour empêcher les utilisateurs ou les rôles d'enregistrer des instances, mettez à jour le profil d'instance pour refuser l'accès à la register commande.

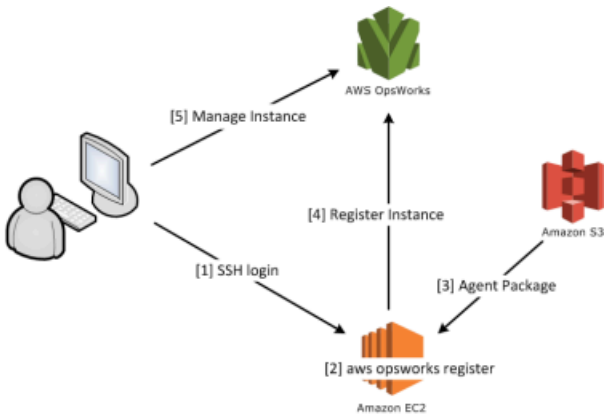
Le processus d'enregistrement installe un agent sur une instance que vous souhaitez gérer à l'aide de AWS OpsWorks Stacks et enregistre l'instance auprès d'une AWS OpsWorks pile que vous spécifiez. Après l'enregistrement de l'instance, celle-ci fait partie de la pile et est gérée par AWS OpsWorks Stacks. Pour de plus amples informations, veuillez consulter Gestion des instances enregistrées.

> **Note**
>
> Bien qu'AWS Tools for PowerShell inclue l'`Register-OpsInstance`applet de commande, qui appelle l'action d'`register`API, nous vous recommandons d'utiliser plutôt le AWS CLI pour exécuter la `register` commande.

Le schéma suivant montre les deux approches pour enregistrer une EC2 instance Amazon. Vous pouvez utiliser les mêmes approches pour enregistrer une instance sur site.



Registering an instance from a workstation



Registering an instance from the instance itself

> **Note**
>
> Vous pouvez utiliser la console AWS OpsWorks Stacks pour gérer une instance enregistrée, mais vous devez exécuter une commande `register` de l'interface de ligne de commande AWS pour enregistrer l'instance. La raison est que le processus d'enregistrement doit être exécuté à partir de l'instance, ce qui ne peut pas être effectué par la console.

Les sections suivantes décrivent la procédure en détail.

Rubriques

- Procédure pas à pas : enregistrer une instance à partir de votre station de travail
- Enregistrement d'Amazon EC2 et d'instances sur site

Procédure pas à pas : enregistrer une instance à partir de votre station de travail

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette fonctionnalité est prise en charge uniquement pour les piles Linux.

Le processus d'enregistrement prend en charge plusieurs scénarios. Cette section vous présente un end-to-end exemple de scénario : comment utiliser votre poste de travail pour enregistrer une EC2 instance Amazon. Les autres scénarios d'enregistrement utilisent une procédure similaire. Pour de plus amples informations, veuillez consulter Enregistrement d'Amazon EC2 et d'instances sur site.

> **Note**
>
> Vous souhaitez généralement enregistrer une EC2 instance Amazon existante. Cependant, vous pouvez juste créer une nouvelle instance et une nouvelle pile pour la procédure pas à pas et les supprimer lorsque vous avez terminé.

Rubriques

- Étape 1 : Créer une pile et une instance
- Étape 2 : Installation et configuration de l'interface AWS CLI
- Étape 3 : enregistrer l'instance auprès de EC2 Register Stack

Étape 1 : Créer une pile et une instance

Pour commencer, vous avez besoin d'une pile et d'une EC2 instance Amazon pour être enregistrées auprès de cette pile.

Pour créer la pile et l'instance

1. Utilisez la console AWS OpsWorks Stacks pour créer une pile nommée **`EC2Register`**. Vous pouvez accepter les valeurs par défaut pour les autres paramètres de la pile.
2. Lancez une nouvelle instance depuis la EC2 console Amazon. Remarques :

   - L'instance doit être dans la même région et le même VPC que la pile.

     Si vous utilisez un VPC, choisissez un sous-réseau public pour cette procédure pas à pas.
   - Si vous devez créer une clé SSH, sauvegardez le fichier de clé privée sur votre station de travail et enregistrez le nom et l'emplacement du fichier.

Si vous utilisez une clé existante, enregistrez le nom et l'emplacement du fichier de clé privée. Vous aurez besoin de ces valeurs plus tard.

- L'instance doit être basée sur l'un des systèmes d'exploitation Linux pris en charge. Par exemple, si votre stack se trouve dans l'ouest des États-Unis (Oregon), vous pouvez `ami-35501205` lancer une instance Ubuntu 14.04 LTS dans cette région.

Sinon, acceptez les valeurs par défaut.

Tandis que l'instance démarre, vous pouvez passer à la section suivante.

Étape 2 : Installation et configuration de l'interface AWS CLI

L'enregistrement est effectué à l'aide de la AWS CLI aws opsworks register commande. Avant d'enregistrer votre première instance, vous devez exécuter la version 1.16.180 AWS CLI ou une version plus récente. Les détails de l'installation dépendent du système d'exploitation de votre station de travail. Pour plus d'informations sur l'installation de AWS CLI, consultez Installation de l'interface de ligne de commande AWS. Pour vérifier la version de l' AWS CLI que vous exécutez, entrez `aws --version` dans une session shell.

> **Note**
>
> Pour empêcher les utilisateurs ou les rôles d'enregistrer des instances, mettez à jour le profil d'instance pour refuser l'accès à la register commande.

Nous vous recommandons vivement de ne pas sauter cette étape, même si vous l'exécutez déjà AWS CLI sur votre poste de travail. Utiliser la version actuelle de AWS CLI est une bonne pratique de sécurité.

Vous devez fournir `register` avec un ensemble d'informations d'identification AWS disposant des autorisations appropriées. La méthode recommandée pour ce faire, afin d'éviter d'installer des informations d'identification directement sur une instance, consiste à enregistrer les instances lancées avec un profil d'instance, puis à ajouter le `--use-instance-profile` commutateur à votre commande. `register` Si vous obtenez des informations d'identification à partir d'un profil d'instance, passez directement à Étape 3 : enregistrer l'instance auprès de EC2 Register Stack dans cette rubrique. Toutefois, si votre instance n'a pas été lancée avec un profil d'instance, vous pouvez créer un utilisateur IAM. La procédure suivante crée un nouvel utilisateur avec les autorisations

appropriées, en installant les informations d'identification de l'utilisateur sur le poste de travail, puis en les transmettant à`register`.

> **Warning**
>
> Les utilisateurs IAM disposent d'informations d'identification à long terme, ce qui présente un risque de sécurité. Pour atténuer ce risque, nous vous recommandons de ne fournir à ces utilisateurs que les autorisations dont ils ont besoin pour effectuer la tâche et de supprimer ces autorisations lorsqu'elles ne sont plus nécessaires.

Pour créer l'utilisateur

1. Dans la console IAM, choisissez Utilisateurs dans le volet de navigation, puis Ajoutez un utilisateur.
2. Ajoutez un utilisateur nommé **`EC2Register`**.
3. Choisissez Suivant.
4. Sur la page Définir les autorisations, choisissez Joindre directement les politiques.
5. Entrez **`OpsWorks`** dans le champ Filtre de politique d'autorisations pour afficher les AWS OpsWorks politiques, sélectionnez l'une des politiques suivantes, puis choisissez Suivant : révision. Cette stratégie accorde à votre utilisateur les autorisations requises pour exécuter `register`.
   - Choisissez `AWSOpsWorksRegisterCLI_EC2` d'autoriser l'utilisateur à enregistrer des EC2 instances utilisant des profils d'instance.
   - Choisissez `AWSOpsWorksRegisterCLI_OnPremises` pour accorder à l'utilisateur les autorisations pour enregistrer des instances sur site.
6. Choisissez Suivant.
7. Sur la page Review (Vérification), choisissez Create user (Créer un utilisateur).
8. Créez maintenant des clés d'accès pour votre utilisateur. Dans le volet de navigation, choisissez Utilisateurs, puis choisissez l'utilisateur pour lequel vous souhaitez créer des clés d'accès.
9. Choisissez l'onglet Informations d'identification de sécurité, puis choisissez Créer une clé d'accès.
10. Choisissez les meilleures pratiques et alternatives relatives aux clés d'accès qui correspondent le mieux à votre tâche.

11. Choisissez Suivant.
12. (Facultatif) entrez un tag pour identifier les clés d'accès.
13. Choisissez Suivant.
14. Choisissez Télécharger le fichier .csv, enregistrez le fichier d'informations d'identification à un emplacement approprié sur votre système, puis cliquez sur OK.

Vous devez fournir les informations d'identification de l'utilisateur IAM à `register`. Cette procédure pas à pas gère la tâche en installant les informations d'identification du EC2 registre dans le `credentials` fichier de votre poste de travail. Pour plus d'informations sur les autres méthodes de gestion des informations d'identification pour le AWS CLI, consultez la section Fichiers de configuration et d'identification.

Pour installer les informations d'identification de l'utilisateur

1. Créez ou ouvrez le fichier `credentials` de votre station de travail. Le fichier se trouve dans `~/.aws/credentials` (Linux, Unix et OS X) ou `C:\Users\`*`User_Name`*`\.aws\credentials` (systèmes Windows).
2. Ajoutez un profil pour l'utilisateur EC2 Register au `credentials` fichier, en utilisant le format suivant.

   ```
   [ec2register]
   aws_access_key_id = access_key_id
   aws_secret_access_key = secret_access_key
   ```

   Remplacez *`access_key_id`* et *`secret_access_key`* par les clés de EC2 registre que vous avez téléchargées précédemment.

Étape 3 : enregistrer l'instance auprès de EC2 Register Stack

Vous êtes maintenant prêt à enregistrer l'instance.

Pour enregistrer l'instance

1. Dans AWS OpsWorks Stacks, revenez à la pile EC2 Register, choisissez Instances dans le volet de navigation, puis sélectionnez Enregistrer une instance.

2. Sélectionnez EC2 Instances, choisissez Next : Select Instances, puis sélectionnez votre instance dans la liste.
3. Choisissez Next : Install AWS CLI, et Next : Register Instances. AWS OpsWorks Stacks utilise automatiquement les informations disponibles, telles que l'ID de pile et l'ID d'instance pour créer un modèle de `register` commande, qui est affiché sur la page Enregistrer les instances. Pour cet exemple, comme `register` devra se connecter à l'instance avec une clé SSH et spécifier explicitement le fichier de clé, définissez J'utilise les clés SSH pour me connecter à mes instances. sur Oui. Le modèle de commande doit se présenter comme suit :

```
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**
>
> Vous devez définir la région sur la région du point de terminaison du service AWS OpsWorks Stacks, et non sur la région de la pile, si la pile se trouve dans une région classique associée au point de terminaison `us-east-1` régional. AWS OpsWorks Stacks détermine la région de la pile à partir de l'ID de la pile.

4. Le modèle de commande contient plusieurs valeurs d'argument spécifiques à l'utilisateur : elles sont indiquées par des crochets et doivent être remplacées par les valeurs appropriées. Copiez le modèle de commande dans un éditeur de texte et modifiez-le comme suit.

> **Important**
>
> L'utilisateur IAM créé au cours du processus d'enregistrement est requis pendant toute la durée de vie d'une instance enregistrée. La suppression de l'utilisateur empêche l'agent AWS OpsWorks Stacks de communiquer avec le service. Pour éviter les problèmes liés à la gestion des instances enregistrées en cas de suppression accidentelle de l'utilisateur, ajoutez le `--use-instance-profile` paramètre à votre `register` commande pour utiliser le profil d'instance intégré de l'instance à la place. L'ajout du `--use-instance-profile` paramètre permet également d'éviter que des erreurs ne se produisent lorsque vous alternez les clés d'accès au AWS compte tous les 90 jours (une bonne pratique recommandée), car cela évite les incohérences entre les clés d'accès disponibles pour l' AWS OpsWorks agent et l'utilisateur IAM requis.

- Remplacez-le *`key file`* par le chemin complet du fichier de clé privée de la paire de EC2 clés Amazon que vous avez enregistrée lors de la création de l'instance.

  Vous pouvez utiliser un chemin relatif, si vous préférez.

- Remplacez *`username`* par le nom d'utilisateur de l'instance.

  Pour cet exemple, le nom d'utilisateur est `ubuntu` pour une instance Ubuntu ou `ec2-user` pour une instance Red Hat Enterprise Linux (RHEL) ou Amazon Linux.

- Ajouter`--use-instance-profile`, qui s'exécute `register` avec le profil d'instance pour éviter les erreurs lors de la rotation des clés ou si l'utilisateur IAM principal est supprimé accidentellement.

Votre commande doit se présenter comme suit :

```
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. Ouvrez une fenêtre terminal sur votre station de travail, collez la commande `register` à partir de votre éditeur et exécutez la commande.

   L'enregistrement prend généralement environ cinq minutes. Une fois l'opération terminée, revenez à la console AWS OpsWorks Stacks et choisissez OK. Ensuite, choisissez Instances dans le panneau de navigation. Votre instance doit être répertoriée sous Instances non attribuées. Vous pouvez ensuite assigner l'instance à une couche ou la laisser où elle est, en fonction de la façon dont vous avez l'intention de gérer l'instance.

6. Lorsque vous avez terminé, arrêtez l'instance, puis supprimez-la à l'aide de la console ou des commandes AWS OpsWorks Stacks. Cela met fin à l' EC2instance Amazon, vous n'avez donc pas à payer de frais supplémentaires.

Enregistrement d'Amazon EC2 et d'instances sur site

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette fonctionnalité est prise en charge uniquement pour les piles Linux.

Cette section explique comment enregistrer une instance Amazon EC2 ou sur site auprès d'une AWS OpsWorks pile Stacks.

Rubriques

- Préparation de l'instance
- Installation et configuration de l' AWS CLI
- Enregistrement de l'instance
- Utilisation de la commande register
- Exemples de commande register
- Stratégies d'enregistrement d'instance

Préparation de l'instance

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette fonctionnalité est prise en charge uniquement pour les piles Linux.

Avant d'enregistrer une instance, vous devez vous assurer qu'elle est compatible avec AWS OpsWorks Stacks. Les détails varient selon que vous enregistrez une EC2 instance sur site ou Amazon.

Instances locales

Une instance sur site doit satisfaire aux critères suivants :

- L'instance doit exécuter l'un des systèmes d'exploitation Linux pris en charge. Bien qu'il soit possible de créer ou d'enregistrer des instances avec d'autres systèmes d'exploitation (tels que CentOS 6). x) qui ont été créés à partir de données personnalisées ou générées par la communauté AMIs, ils ne sont pas officiellement pris en charge.

  Vous devez installer le package `libyaml` sur l'instance. Pour les instances Ubuntu, le package se nomme `libyaml-0-2`. Pour les instances CentOS et Red Hat Enterprise Linux, le package se nomme `libyaml`.
- L'instance doit avoir un type d'instance pris en charge (parfois appelé taille d'instance). Les types d'instance pris en charge peuvent varier selon le système d'exploitation et si votre pile se trouve dans un VPC. Pour obtenir la liste des types d'instances pris en charge, consultez les valeurs de la liste déroulante Taille affichées dans la console AWS OpsWorks Stacks lorsque vous essayez de créer une nouvelle instance dans votre pile cible. Si un type d'instance apparaît en grisé et ne peut pas être créé dans votre pile cible, vous ne pouvez pas enregistrer d'instance de ce type.
- L'instance doit disposer d'un accès Internet lui permettant de communiquer avec le point de terminaison du service AWS OpsWorks Stacks,`opsworks.us-east-1.amazonaws.com (HTTPS)`. L'instance doit également prendre en charge les connexions sortantes vers les ressources AWS telles qu'Amazon S3.
- Si vous avez l'intention d'enregistrer l'instance à partir d'une station de travail distincte, l'instance enregistrée doit prendre en charge la connexion SSH à partir de la station de travail.

  La connexion SSH n'est pas obligatoire si vous exécutez la commande d'enregistrement à partir de l'instance.
- La clé AWS d'accès est utilisée pour l'authentification entre l' AWS OpsWorks agent et le service AWS OpsWorks Stacks. Si vous alternez les clés d'accès comme recommandé tous les 90 jours,

mettez à jour l' AWS OpsWorks agent manuellement pour qu'il utilise la nouvelle clé. Sur un ordinateur ou une instance sur site, modifiez le `/etc/aws/opsworks/instance-agent.yml` fichier avec la nouvelle clé d'accès et la nouvelle clé secrète. La commande suivante illustre la clé d'accès et la clé secrète dans ce fichier. Un agent utilisant d'anciennes clés peut entraîner des erreurs.

```
cat /etc/aws/opsworks/instance-agent.yml | egrep "access_key|secret_key"
:access_key_id: AKIAIOSFODNN7EXAMPLE
:secret_access_key: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

EC2Instances Amazon

Une EC2 instance Amazon doit répondre aux critères suivants :

- L'AMI doit être basée sur l'un des systèmes d'exploitation Linux pris en charge. Pour obtenir la liste actuelle, consultez AWS OpsWorks Systèmes d'exploitation Stacks.

  Pour de plus amples informations, veuillez consulter Utilisation de Custom AMIs.

  Si l'instance est basée sur une AMI personnalisée qui dérive d'une AMI standard supportée, ou si l'instance contient une installation minimale, vous devez installer le package `libyaml` sur l'instance. Pour les instances Ubuntu, le package se nomme `libyaml-0-2`. Pour les instances Amazon Linux et Red Hat Enterprise Linux, le package est nommé`libyaml`.
- L'instance doit avoir un type d'instance pris en charge (parfois appelé taille d'instance). Les types d'instance pris en charge peuvent varier selon le système d'exploitation et si votre pile se trouve dans un VPC. Pour obtenir la liste des types d'instances pris en charge, consultez les valeurs de la liste déroulante Taille affichées dans la console AWS OpsWorks Stacks lorsque vous essayez de créer une nouvelle instance dans votre pile cible. Si un type d'instance apparaît en grisé et ne peut pas être créé dans votre pile cible, vous ne pouvez pas enregistrer d'instance de ce type, non plus.
- L'instance doit être dans l'état `running`.
- L'instance ne doit pas faire partie d'un groupe Auto scaling.

  Pour plus d'informations, consultez Detach EC2 Instances From Your Auto Scaling Group.
- L'instance peut faire partie d'un VPC, mais elle doit se trouver dans le même VPC que la pile et le VPC doit être configuré pour fonctionner correctement avec Stacks. AWS OpsWorks
- Les instances Spot ne sont pas prises en charge, car elles ne fonctionnent pas avec la réparation automatique.

Lorsque vous enregistrez une EC2 instance Amazon, AWS OpsWorks Stacks ne modifie pas les groupes de sécurité ou les règles de l'instance. Assurez-vous que les règles du groupe de sécurité de l'instance répondent aux exigences AWS OpsWorks Stacks suivantes.

Règles d'entrée

Les règles d'entrée doivent autoriser les actions suivantes :

- Connexion SSH.
- Trafic à partir des couches appropriées.

  Par exemple, un serveur de base de données autorise généralement le trafic entrant à partir des couches serveur d'applications de la pile.
- Trafic vers les ports appropriés.

  Par exemple, les instances serveur d'applications autorisent généralement tout le trafic entrant sur les ports 80 (HTTP) et 443 (HTTPS).

Règles de sortie

Les règles d'entrée doivent autoriser les actions suivantes :

- Trafic vers le service AWS OpsWorks Stacks depuis les applications exécutées sur l'instance.
- Trafic permettant d'accéder aux ressources AWS telles qu'Amazon S3 à partir d'applications utilisant l'API AWS.

Une approche courante consiste à ne pas spécifier de règles de sortie, ce qui n'impose aucune restriction sur le trafic sortant.

Installation et configuration de l' AWS CLI

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Avant d'enregistrer votre première instance, vous devez exécuter la version 1.16.180 AWS CLI ou une version plus récente sur l'ordinateur à partir duquel vous l'exécutez. `register` Les détails de l'installation dépendent du système d'exploitation de votre station de travail. Pour plus d'informations sur l'installation de AWS CLI, consultez les sections Installation de l'interface de ligne de commande AWS et Configuration de l'interface de ligne de commande AWS. Pour vérifier la version de l' AWS CLI que vous exécutez, entrez `aws --version` dans une session shell.

> **Note**
>
> Bien qu'AWS Tools for PowerShell inclue l'`Register-OpsInstance`applet de commande, qui appelle l'action d'`register`API, nous vous recommandons d'utiliser plutôt le AWS CLI pour exécuter la `register` commande.

Vous devez exécuter `register` avec les autorisations appropriées. Vous pouvez obtenir des autorisations en utilisant un rôle IAM, ou de manière moins optimale, en installant les informations d'identification utilisateur avec les autorisations appropriées sur le poste de travail ou l'instance à enregistrer. Vous pouvez ensuite exécuter `register` avec ces informations d'identification, comme décrit ci-après. Spécifiez les autorisations en associant une politique IAM à l'utilisateur ou au rôle. En `register` effet, vous utilisez les `AWSOpsWorksRegisterCLI_OnPremises` politiques `AWSOpsWorksRegisterCLI_EC2` ou, qui accordent des autorisations pour enregistrer des instances Amazon EC2 ou sur site, respectivement.

> **Note**
>
> Si vous exécutez `register` sur une EC2 instance Amazon, vous devez idéalement utiliser un rôle IAM pour fournir des informations d'identification. Pour plus d'informations sur la façon d'associer un rôle IAM à une instance existante, consultez Attacher un rôle IAM à une instance ou Remplacer un rôle IAM dans le guide de l'utilisateur Amazon EC2 .

Pour voir des exemples d'extraits de stratégies `AWSOpsWorksRegisterCLI_EC2` et `AWSOpsWorksRegisterCLI_OnPremises`, consultez Stratégies d'enregistrement d'instance. Pour plus d'informations sur la création et la gestion des informations d'identification AWS, consultez Informations d'identification de sécurité AWS.

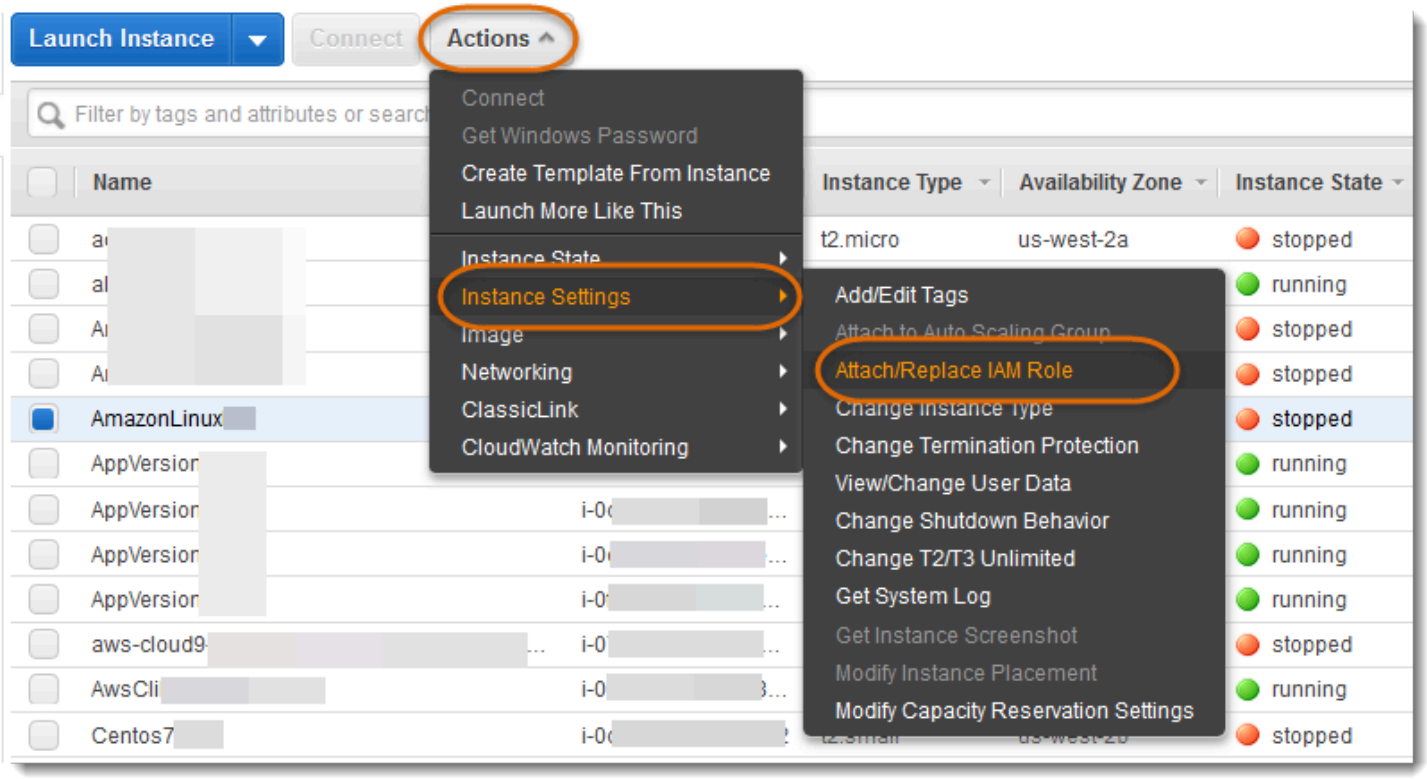
Rubriques

- Utilisation d'un rôle IAM

- Utilisation des informations d'identification installées

Utilisation d'un rôle IAM

Si vous exécutez la commande depuis l' EC2 instance Amazon que vous souhaitez enregistrer, la stratégie préférée pour fournir des informations d'identification `register` consiste à utiliser un rôle IAM associé à la `AWSOpsWorksRegisterCLI_EC2` politique ou à des autorisations équivalentes. Cette approche vous permet d'éviter l'installation de vos informations d'identification sur l'instance. Pour ce faire, vous pouvez utiliser la commande Attacher/Remplacer le rôle IAM dans la EC2 console, comme illustré dans l'image suivante.



Pour plus d'informations sur la façon d'associer un rôle IAM à une instance existante, consultez Attacher un rôle IAM à une instance ou Remplacer un rôle IAM dans le guide de l'utilisateur Amazon EC2 . Pour les instances lancées avec un profil d'instance (recommandé), ajoutez le `--use-instance-profile` commutateur à votre commande `register` pour fournir vos informations d'identification, ne pas utiliser le paramètre `--profile`.

Si l'instance est en cours d'exécution et a un rôle, vous pouvez accorder les autorisations requises en associant la stratégie `AWSOpsWorksRegisterCLI_EC2` au rôle. Le rôle fournit à l'instance un ensemble d'informations d'identification par défaut. Tant que vous n'avez pas installé d'informations

d'identification sur l'instance, `register` assume automatiquement le rôle et s'exécute avec ses autorisations.

> **Important**
>
> Nous vous recommandons de ne pas installer d'informations d'identification sur l'instance. En plus de créer un risque de sécurité, le rôle de l'instance se situe à la fin de la chaîne de fournisseurs par défaut qu'elle AWS CLI utilise pour localiser les informations d'identification par défaut. Les informations d'identification installées peuvent avoir priorité sur le rôle et, par conséquent, `register` pourrait ne pas avoir les autorisations requises. Pour plus d'informations, consultez Démarrer avec le AWS CLI.

Si une instance en cours d'exécution n'a pas de rôle, vous devez installer les informations d'identification avec les autorisations requises sur l'instance, comme décrit dans Utilisation des informations d'identification installées. Il est recommandé, plus facile et plus fiable d'utiliser les instances lancées avec un profil d'instance.

Utilisation des informations d'identification installées

Il existe plusieurs manières d'installer les informations d'identification utilisateur sur un système et de les fournir à une AWS CLI commande. Ce qui suit décrit une approche qui n'est plus recommandée, mais qui peut être utilisée si vous enregistrez des EC2 instances lancées sans profil d'instance. Vous pouvez aussi utiliser les informations d'identification d'un utilisateur existant, aussi longtemps que les stratégies attachées accordent les autorisations requises. Pour plus d'informations, y compris la description des autres façons d'installer les informations d'identification, consultez Fichiers de configuration et d'informations d'identification.

Pour utiliser les informations d'identification installées

1. Créez un utilisateur IAM et enregistrez l'ID de la clé d'accès et la clé d'accès secrète dans un emplacement sécurisé.

   > **Warning**
   >
   > Les utilisateurs IAM disposent d'informations d'identification à long terme, ce qui présente un risque de sécurité. Pour atténuer ce risque, nous vous recommandons de ne

fournir à ces utilisateurs que les autorisations dont ils ont besoin pour effectuer la tâche et de supprimer ces autorisations lorsqu'elles ne sont plus nécessaires.

2. Attachez la AWSOpsWorksRegisterCLI_OnPremises politique à l'utilisateur. Si vous préférez, vous pouvez attacher une stratégie qui accorde des autorisations de plus grande ampleur, à partir du moment où elle comporte les autorisations `AWSOpsWorksRegisterCLI_OnPremises`.
3. Créez un profil pour l'utilisateur dans le fichier `credentials` du système. Le fichier se trouve dans `~/.aws/credentials` (Linux, Unix et OS X) ou `C:\Users\`*`User_Name`*`\.aws\credentials` (systèmes Windows). Le fichier contient un ou plusieurs profils au format suivant, chacun contenant l'ID de clé d'accès et la clé d'accès secrète d'un utilisateur.

```
[profile_name]
aws_access_key_id = access_key_id
aws_secret_access_key = secret_access_key
```

   Remplacez les informations d'identification IAM que vous avez enregistrées précédemment par les *`secret_access_key`* valeurs *`access_key_id`* et. Vous pouvez spécifier le nom de votre choix pour un nom de profil, avec deux limites : le nom doit être unique et le profil par défaut doit se nommer `default`. Vous pouvez aussi utiliser un profil existant, aussi longtemps qu'il possède les permissions requises.
4. Utilisez le paramètre `register` de la commande `--profile` pour spécifier le nom du profil. La commande `register` exécute les autorisations accordées aux informations d'identification associées.

   Vous pouvez également ignorer `--profile`. Dans ce cas, `register` s'exécute avec les informations d'identification par défaut. Sachez que comme il ne s'agit pas nécessairement des informations d'identification du profil par défaut, vous devez vous assurer que les informations d'identification par défaut disposent des autorisations requises. Pour plus d'informations sur la façon dont les AWS CLI informations d'identification par défaut sont déterminées, consultez Configuration de l'interface de ligne de commande AWS.

Enregistrement de l'instance

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette fonctionnalité est prise en charge uniquement pour les piles Linux.

Vous enregistrez une instance en exécutant la commande AWS CLI `register` à partir de votre station de travail ou de l'instance. La façon la plus simple de gérer l'opération consiste à utiliser l'Assistant d'enregistrement de la console AWS OpsWorks Stacks, ce qui simplifie le processus de construction de la chaîne de commande. Une fois que vous êtes familiarisé avec la procédure d'enregistrement, vous pouvez ignorer l'Assistant si vous le souhaitez et exécuter la commande `register`.

Ce qui suit décrit comment utiliser l'Assistant d'enregistrement pour enregistrer une instance auprès d'une pile existante.

> **Note**
>
> Pour enregistrer une instance auprès d'une nouvelle pile, vous pouvez le faire en choisissant Enregistrer les instances dans le tableau de bord AWS OpsWorks Stacks. Un Assistant, identique à celui des piles existantes, à l'exception d'une page supplémentaire qui configure la nouvelle pile, démarre alors.

Pour utiliser l'Assistant d'enregistrement pour enregistrer une instance

1. Dans la console AWS OpsWorks Stacks, créez une pile ou ouvrez une pile existante.

2. Dans le volet de navigation, sélectionnez Instances, puis choisissez register an instance (enregistrer une instance).
3. Sur la page Choisissez un type d'instance, indiquez si vous souhaitez enregistrer une instance Amazon EC2 ou une instance sur site :
   - Si vous enregistrez une EC2 instance Amazon, choisissez Next : Select Instances.
   - Si vous enregistrez une instance sur site, choisissez Next : Install AWS CLI, puis passez à l'étape 5.
4. Si vous enregistrez une EC2 instance Amazon, ouvrez la page Select Instances pour sélectionner l'instance à enregistrer. AWS OpsWorks Stacks collecte les informations nécessaires à la création de la commande. Lorsque vous avez terminé, choisissez Next:Install AWS CLI (Suivant : Installer l'interface de ligne de commande).
5. L'instance sur laquelle vous comptez exécuter `register` doit exécuter la version 1.16.180 AWS CLI ou une version ultérieure. Pour installer ou mettre à jour l' AWS CLI, la page de l'Assistant d'enregistrement fournit des liens vers les instructions d'installation et de configuration. Après avoir vérifié l'installation de l' AWS CLI , spécifiez si vous exécutez la commande à partir de l'instance à enregistrer ou d'une station de travail distincte, puis choisissez Next: Register Instances (Suivant : Enregistrer des instances).
6. La page Register Instances (Enregistrer des instances) affiche un modèle pour une chaîne de commande `register` qui intègre vos options choisies. Par exemple, si vous enregistrez une EC2 instance Amazon à partir d'un poste de travail distinct, le modèle par défaut ressemble au suivant.

```
aws opsworks register --infrastructure-class ec2 --region us-west-2
  --stack-id 247be7ea-3551-4177-9524-1ff804f453e3 --ssh-username [username] i-
f1245d10
```

> **Important**
>
> L'utilisateur IAM créé lors du processus d'enregistrement est requis pendant toute la durée de vie d'une instance enregistrée. La suppression de l'utilisateur empêche l'agent AWS OpsWorks Stacks de communiquer avec le service. Pour éviter les problèmes liés à la gestion des instances enregistrées en cas de suppression accidentelle de l'utilisateur, ajoutez le `--use-instance-profile` paramètre à votre `register` commande pour utiliser le profil d'instance intégré de l'instance à la place. L'ajout du `--use-instance-profile` paramètre permet également d'éviter que des erreurs ne se

produisent lorsque vous alternez les clés d'accès au AWS compte tous les 90 jours (une bonne pratique recommandée), car cela évite les incohérences entre les clés d'accès disponibles pour l' AWS OpsWorks agent et l'utilisateur IAM requis.

Si vous définissez I use SSH keys sur Oui, AWS OpsWorks Stacks ajoute l'`--ssh-private-key`argument à la chaîne, que vous pouvez utiliser pour spécifier un fichier de clé SSH privé.

> **Note**
>
> Si vous souhaitez vous `register` connecter avec un mot de passe, réglez I use SSH keys sur Non. Lorsque vous exécutez`register`, le mot de passe vous est demandé.

Copiez cette chaîne dans un éditeur de texte et modifiez-la si nécessaire. Remarques :

- Le texte entre crochets représente les informations que vous devez fournir, telles que l'emplacement de votre fichier de clé SSH.
- Le modèle présume que vous exécutez `register` avec les informations d'identification AWS par défaut. Dans le cas contraire, ajoutez un argument `--profile` à la chaîne de commande, puis spécifiez le nom du profil d'informations d'identification que vous souhaitez utiliser.

Pour les autres scénarios, vous devrez peut-être modifier la commande ultérieurement. Pour obtenir une explication des arguments `register` disponibles et des autres moyens de construire la chaîne de commande, consultez Utilisation de la commande `register`. Vous pouvez aussi afficher la documentation de la commande en exécutant `aws opsworks help register` à partir de la ligne de commande. Pour obtenir des exemples de chaînes de commande, consultez Exemples de commande register.

7. Une fois que vous avez terminé de modifier la chaîne de commande, ouvrez une fenêtre de terminal sur votre station de travail, ou utilisez SSH pour vous connecter à l'instance et exécutez la commande. L'opération complète nécessite généralement cinq minutes environ, pendant lesquelles l'instance se trouve dans l'état Registering (Enregistrement en cours).
8. Une fois l'opération terminée, choisissez Done (Terminé). L'instance est à présent dans l'état Registered (Enregistré) et apparaît comme instance non assignée sur la page Instances de la pile.

La commande `register` exécute les opérations suivantes :

1. Si `register` s'exécute sur une station de travail, la commande utilise d'abord SSH pour se connecter à l'instance à enregistrer.

   Le reste du processus se déroule sur l'instance et demeure le même, quel que soit l'endroit où vous avez exécuté la commande.
2. Télécharge le package de l'agent AWS OpsWorks Stacks depuis Amazon S3.
3. Décompresse et installe l'agent et ses dépendances, telles que le Kit de développement logiciel (SDK) AWS pour Ruby.
4. Crée les éléments suivants :
   - Un utilisateur IAM qui démarre l'agent avec le service AWS OpsWorks Stacks pour garantir une communication sécurisée.

     Les autorisations de l'utilisateur ne permettent que l'action `opsworks:RegisterInstance` et expirent après 15 minutes.
   - Un groupe IAM pour la pile, qui contient les utilisateurs des instances enregistrées.
5. Crée une paire de clés RSA et envoie la clé publique à AWS OpsWorks Stacks.

   Cette paire de clés est utilisée pour chiffrer les communications entre l'agent et AWS OpsWorks Stacks.
6. Enregistre l'instance auprès de AWS OpsWorks Stacks. La pile exécute ensuite un ensemble de recettes de configuration initiale pour configurer l'instance, ce qui inclut les actions suivantes :
   - Remplacement du fichier hôtes de l'instance.

     En enregistrant l'instance, vous avez transféré la gestion des utilisateurs à AWS OpsWorks Stacks, qui doit disposer de son propre fichier d'hôtes pour contrôler les autorisations de connexion SSH.
   - Pour les EC2 instances Amazon, la configuration initiale inclut également l'enregistrement de tous les volumes Amazon EBS ou adresses IP Elastic attachés auprès de la pile.

     Vous devez vous assurer que les volumes Amazon EBS ne sont pas montés sur des points de montage réservés, y compris `/var/www` les points de montage réservés par les couches de l'instance. Pour plus d'informations sur la gestion des ressources d'une pile, consultez Gestion des ressources. Pour plus d'informations sur les points de montage d'une couche, consultez AWS OpsWorks Référence de la couche Stacks.

Pour obtenir la description complète des modifications de la configuration initiale, consultez Modifications de la configuration d'installation initiale.

> **Note**
>
> La configuration initiale ne met pas à jour le système d'exploitation d'une instance enregistrée ; vous devez gérer cette tâche vous-même. Pour de plus amples informations, veuillez consulter Gestion des mises à jour de sécurité.

Utilisation de la commande **`register`**

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette fonctionnalité est prise en charge uniquement pour les piles Linux.

Pour enregistrer une instance, assurez-vous que vous exécutez au moins la version 1.16.180 de AWS CLI. Ce qui suit montre la syntaxe générale de la commande `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]
```

Les arguments suivants sont communs à toutes les AWS CLI commandes.

`--profile`

(Facultatif) Nom de profil des informations d'identification. Si vous omettez cet argument, la commande s'exécute avec vos informations d'identification par défaut. Pour plus d'informations sur la façon dont les AWS CLI informations d'identification par défaut sont déterminées, consultez Configuration de l'interface de ligne de commande AWS.

`--region`

(Facultatif) Région du point de terminaison du service AWS OpsWorks Stacks. Ne définissez pas `--region` la région de la pile. AWS OpsWorks Stacks détermine automatiquement la région de la pile à partir de l'ID de la pile.

> **Note**
>
> Si votre région par défaut est déjà définie, vous pouvez omettre cet argument. Pour plus d'informations sur la manière de spécifier une région par défaut, consultez Configuration de l'interface de ligne de commande AWS.

Utilisez les arguments suivants pour Amazon EC2 et les instances sur site.

`--infrastructure-class`

(Obligatoire) Ce paramètre doit être défini sur `ec2` ou `on-premises` pour indiquer si vous enregistrez une instance Amazon EC2 ou sur site, respectivement.

`--stack-id`

(Obligatoire) ID de la pile auprès de laquelle l'instance doit être enregistrée.

> **Note**
>
> Pour trouver un ID de pile, sur la page Pile, choisissez Paramètres. L'ID de pile est étiqueté OpsWorks ID. Il s'agit d'un GUID qui ressemble à quelque chose comme `ead21bce6-7623-47f1-bf9d-af2affad8907`.

Arguments de connexion SSH

Utilisez les arguments suivants pour spécifier comment `register` doit se connecter à l'instance.

`--local`

(Facultatif) Utilisez cet argument pour enregistrer l'instance sur laquelle vous exécutez la commande.

Dans ce cas, `register` n'a pas besoin de se connecter à l'instance.

`--ssh-private-key` et **`--ssh-username`**

(Facultatif) Utilisez ces arguments si vous enregistrez l'instance à partir d'une station de travail distincte et que vous voulez spécifier explicitement le nom d'utilisateur ou le fichier de clé privée.

- `--ssh-username`— Utilisez cet argument pour spécifier un nom d'utilisateur SSH.

  Si vous omettez `--ssh-username`, `ssh` utilise le nom d'utilisateur par défaut.
- `--ssh-private-key`— Utilisez cet argument pour spécifier explicitement un fichier de clé privée.

  Si vous omettez `--ssh-private-key`, `ssh` tente de se connecter à l'aide des techniques d'authentification ne nécessitant pas un mot de passe, y compris l'utilisation de la clé privée par défaut. Si aucune de ces techniques n'est prise en charge, `ssh` interroge votre mot de passe. Pour plus d'informations sur la façon dont `ssh` gère l'authentification, consultez The Secure Shell (SSH) Authentication Protocol.

`--override-ssh`

(Facultatif) Utilisez cet argument si vous enregistrez l'instance à partir d'une station de travail distincte et que vous souhaitez spécifier une chaîne de commande `ssh` personnalisée. La commande `register` utilise cette chaîne de commande pour se connecter à l'instance enregistrée.

Pour plus d'informations sur `ssh`, consultez SSH.

`--override-hostname`

(Facultatif) Spécifie un nom d'hôte pour l'instance, qui est utilisé uniquement par AWS OpsWorks Stacks. La valeur par défaut est le nom d'hôte de l'instance.

`--debug`

(Facultatif) Fournit les informations de débogage en cas d'échec du processus d'enregistrement. Pour plus d'informations sur le dépannage, consultez Dépannage de l'enregistrement des instances.

`--use-instance-profile`

(Facultatif, mais fortement recommandé pour les EC2 instances Amazon) Permet à la `register` commande d'utiliser un profil d'instance attaché, au lieu de créer un utilisateur IAM. L'ajout de ce paramètre permet d'éviter que des erreurs ne se produisent si vous essayez de gérer une instance enregistrée lorsque l'utilisateur IAM a été supprimé accidentellement.

> **Important**
>
> L'utilisateur IAM créé lors du processus d'enregistrement est requis pendant toute la durée de vie d'une instance enregistrée. La suppression de l'utilisateur empêche l'agent AWS OpsWorks Stacks de communiquer avec le service. Pour éviter les problèmes liés à la gestion des instances enregistrées en cas de suppression accidentelle de l'utilisateur, ajoutez le `--use-instance-profile` paramètre à votre `register` commande pour utiliser le profil d'instance intégré de l'instance à la place. L'ajout du `--use-instance-profile` paramètre empêche également les erreurs de se produire lorsque vous alternez les clés d'accès au AWS compte tous les 90 jours (une bonne pratique recommandée), car cela permet d'éviter les incohérences entre les clés d'accès disponibles pour l' AWS OpsWorks agent et celles de l'utilisateur requis.

Cible

(Conditionnel) Si vous exécutez cette commande à partir d'une station de travail, la valeur finale de la chaîne de commande spécifie la cible d'enregistrement de l'une des façons suivantes.

- Adresse IP publique de l'instance.
- Nom d'hôte de l'instance.

- Pour les EC2 instances Amazon, l'ID de l'instance.

  AWS OpsWorks Stacks utilise l'ID de l'instance pour obtenir la configuration de l'instance, y compris l'adresse IP publique de l'instance. Par défaut, AWS OpsWorks Stacks utilise cette adresse pour créer la chaîne de `ssh` commande qu'il utilise pour se connecter à l'instance. Si vous avez besoin de vous connecter à une adresse IP privée, vous devez utiliser `--override-ssh` pour fournir une chaîne de commande personnalisée. Pour obtenir un exemple, consultez Enregistrer une instance locale à partir d'une station de travail.

  > **Note**
  >
  > Si vous spécifiez un nom d'hôte, `ssh` dépend du serveur DNS pour résoudre le nom en une instance particulière. Si vous n'êtes pas certain que le nom d'hôte soit unique, utilisez `ssh` pour vérifier que le nom d'hôte se résout en l'instance correcte.

  Si vous exécutez cette commande à partir de l'instance à enregistrer, omettez l'identifiant d'instance et utilisez à la place l'argument `--local`.

Les arguments suivants concernent uniquement les instances locales.

`--override-public-ip`

(Facultatif) AWS OpsWorks Stacks affiche l'adresse spécifiée en tant qu'adresse IP publique de l'instance. Il ne change pas l'adresse IP publique de l'instance. Toutefois, si un utilisateur utilise la console pour se connecter à l'instance, par exemple en choisissant l'adresse sur la page Instances, AWS OpsWorks Stacks utilise l'adresse spécifiée. AWS OpsWorks Stacks détermine automatiquement la valeur par défaut de l'argument.

`--override-private-ip`

(Facultatif) AWS OpsWorks Stacks affiche l'adresse spécifiée comme adresse IP privée de l'instance. Cela ne modifie pas l'adresse IP privée de l'instance. AWS OpsWorks Stacks détermine automatiquement la valeur par défaut de l'argument.

Exemples de commande register

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette fonctionnalité est prise en charge uniquement pour les piles Linux.

Cette section contient des exemples de chaînes de commande `register`.

Enregistrer une EC2 instance Amazon depuis un poste de travail

L'exemple suivant enregistre une EC2 instance Amazon depuis un poste de travail. La chaîne de commande utilise les informations d'identification par défaut et identifie l'instance par son identifiant d' EC2 instance Amazon. Vous pouvez utiliser cet exemple pour les instances sur site en passant `ec2` à`on-premises`.

```
aws opsworks register \
  --region us-west-2 \
  --use-instance-profile \
  --infrastructure-class ec2 \
  --stack-id ad21bce6-7623-47f1-bf9d-af2affad8907 \
  --ssh-user-name my-sshusername \
  --ssh-private-key "./keys/mykeys.pem" \
  i-2422b9c5
```

Enregistrer une instance locale à partir d'une station de travail

L'exemple suivant enregistre une instance locale à partir d'une station de travail distincte. La chaîne de commande utilise les informations d'identification par défaut et se connecte à l'instance

avec la chaîne de commande `ssh` spécifiée. Si votre instance nécessite un mot de passe, `register` vous invite à le saisir. Vous pouvez utiliser l'exemple pour les EC2 instances Amazon en passant `on-premises` à`ec2`.

```
aws opsworks register \
  --region us-west-2 \
  --infrastructure-class on-premises \
  --stack-id ad21bce6-7623-47f1-bf9d-af2affad8907 \
  --override-ssh "ssh your-user@192.0.2.0"
```

> **Note**
>
> Vous pouvez l'utiliser `--override-ssh` pour spécifier n'importe quelle chaîne de commande SSH personnalisée. AWS OpsWorks Stacks utilise ensuite la chaîne spécifiée pour se connecter à l'instance au lieu de construire une chaîne de commande. Pour obtenir un autre exemple, consultez Enregistrer une instance à l'aide d'une chaîne de commande SSH personnalisée.

Enregistrer une instance à l'aide d'une chaîne de commande SSH personnalisée

L'exemple suivant enregistre une instance locale à partir d'un poste de travail et utilise l'`--override-ssh`argument pour spécifier une commande SSH personnalisée qui `register` permet de se connecter à l'instance. Cet exemple utilise `sshpass` pour se connecter avec un nom d'utilisateur et un mot de passe, mais vous pouvez spécifier n'importe quelle chaîne de commande `ssh` valide.

```
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"
```

Enregistrer une instance en exécutant **`register`** à partir de l'instance

L'exemple suivant montre comment enregistrer une EC2 instance Amazon en l'exécutant `register` à partir de l'instance elle-même. La chaîne de commande dépend des informations d'identification par défaut pour ses autorisations. Pour utiliser l'exemple pour une instance locale, passez `--infrastructure-class` à`on-premises`.

```
aws opsworks register \
  --region us-west-2 \
  --infrastructure-class ec2 \
  --stack-id ad21bce6-7623-47f1-bf9d-af2affad8907 \
  --local
```

Enregistrer une instance avec une adresse IP privée

Par défaut, `register` utilise l'adresse IP publique de l'instance pour se connecter à l'instance. Pour enregistrer une instance avec une adresse IP privée, telle qu'une instance du sous-réseau privé d'un VPC, vous devez utiliser `--override-ssh` pour spécifier une chaîne de commande `ssh` personnalisée.

```
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
```

Stratégies d'enregistrement d'instance

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les `AWSOpsWorksRegisterCLI_OnPremises` politiques `AWSOpsWorksRegisterCLI_EC2` et fournissent les autorisations appropriées pour les instances d'enregistrement EC2 et sur site, respectivement. Vous ajoutez `AWSOpsWorksRegisterCLI_EC2` à votre utilisateur IAM pour enregistrer des EC2 instances, mais vous ajoutez `AWSOpsWorksRegisterCLI_OnPremises` à votre utilisateur pour enregistrer des instances sur site. Pour utiliser ces politiques, vous devez exécuter au moins la version 1.16.180 AWS CLI ou une version plus récente.

La stratégie **AWSOpsWorksRegisterCLI_EC2**

Ajoutez `AWSOpsWorksRegisterCLI_EC2` à votre utilisateur pour enregistrer EC2 des instances. Vous devez utiliser ce profil si vous prévoyez d'enregistrer uniquement EC2 des instances. Lorsque vous utilisez cette politique, les autorisations sont fournies par le profil d' EC2instance de l'instance.

```
{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "opsworks:AssignInstance",
            "opsworks:CreateLayer",
            "opsworks:DeregisterInstance",
            "opsworks:DescribeInstances",
            "opsworks:DescribeStackProvisioningParameters",
            "opsworks:DescribeStacks",
            "opsworks:UnassignInstance"
          ],
          "Resource": [
            "*"
          ]
        },
        {
          "Effect": "Allow",
          "Action": [
            "ec2:DescribeInstances"
          ],
          "Resource": [
            "*"
          ]
        }
      ]
    }
```

La stratégie **AWSOpsWorksRegisterCLI_OnPremises**

Ajoutez `AWSOpsWorksRegisterCLI_OnPremises` à votre utilisateur pour enregistrer des instances sur site. Cette politique inclut les autorisations IAM, par exemple`AttachUserPolicy`, mais les ressources sur lesquelles ces autorisations s'appliquent sont limitées.

```
    {
```

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

(Obsolète) La stratégie **`AWSOpsWorksRegisterCLI`**

> **Important**
>
> La stratégie `AWSOpsWorksRegisterCLI` est obsolète et ne peut pas être utilisée pour enregistrer de nouvelles instances. Elle est disponible uniquement pour la rétrocompatibilité sur des instances qui ont déjà été enregistrées. La `AWSOpsWorksRegisterCLI` politique inclut de nombreuses autorisations IAM`CreateUser`, notamment`PutUserPolicy`, et`AddUserToGroup`. Comme il s'agit d'autorisations de niveau administrateur, vous ne devez attribuer la stratégie `AWSOpsWorksRegisterCLI` qu'aux utilisateurs administratifs approuvés.

Gestion des instances enregistrées

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette fonctionnalité est prise en charge uniquement pour les piles Linux.

Lorsque vous enregistrez une instance, elle devient une instance AWS OpsWorks Stacks et vous pouvez la gérer de la même manière que les instances créées avec AWS OpsWorks Stacks. Il existe deux différences principales :

- Les instances enregistrées n'ont pas à être attribuées à une couche.
- Vous pouvez annuler l'enregistrement d'une instance et la retourner sous votre contrôle direct.

Une fois que vous avez enregistré une instance, elle passe à l'état Enregistré. AWS OpsWorks Stacks fournit les fonctionnalités de gestion suivantes à toutes les instances enregistrées :

- Health checks — AWS OpsWorks Stacks surveille l'agent pour évaluer si l'instance continue de fonctionner.

  Si le bilan de santé d'une instance échoue, AWS OpsWorks Stacks répare automatiquement les EC2 instances Amazon enregistrées et change le statut des instances locales enregistrées en. `connection lost`
- CloudWatch surveillance : CloudWatch la surveillance est activée pour l'instance enregistrée.

  Vous pouvez surveiller les métriques, telles que l'utilisation de l'UC et la mémoire disponible, et le cas échéant recevoir une notification si une métrique dépasse un seuil défini.
- Gestion des utilisateurs — AWS OpsWorks Stacks fournit un moyen simple de spécifier quels utilisateurs peuvent accéder à l'instance et quelles opérations ils sont autorisés à effectuer. Pour de plus amples informations, veuillez consulter Gestion des autorisations utilisateur.
- Exécution de recettes — Vous pouvez utiliser la commande Execute Recipes stack pour exécuter des recettes Chef sur l'instance.
- Mises à jour du système d'exploitation : vous pouvez utiliser la commande Update Dependencies stack pour mettre à jour le système d'exploitation de l'instance.

Pour tirer pleinement parti de la fonctionnalité de gestion des AWS OpsWorks Stacks, vous pouvez attribuer l'instance à une couche. Pour de plus amples informations, veuillez consulter Attribution d'une instance enregistrée à une couche.

Il existe des différences entre la façon dont AWS OpsWorks Stacks gère Amazon EC2 et les instances sur site.

EC2 Instances Amazon

- Si vous arrêtez une EC2 instance Amazon enregistrée, AWS OpsWorks Stacks met fin aux instances sauvegardées par le stockage d'instance et arrête les instances soutenues par Amazon EBS.

  Comme l'instance est toujours enregistrée auprès de la pile et assignée à ses couches, vous pouvez la redémarrer si nécessaire. Vous devez annuler l'enregistrement d'une instance pour la supprimer d'une pile, explicitement ou en supprimant l'instance, ce qui annule l'enregistrement l'automatiquement.
- Si vous redémarrez une EC2 instance Amazon enregistrée ou si l'instance échoue et est réparée automatiquement, le résultat est le même que l'arrêt et le redémarrage de l'instance à l'aide d'Amazon. EC2 Notez ces différences :
  - Instances basées sur le stockage d'instances : AWS OpsWorks Stacks démarre une nouvelle instance avec la même AMI.

    Notez que AWS OpsWorks Stacks n'a connaissance d'aucune opération que vous avez effectuée sur l'instance avant son enregistrement, comme l'installation de packages logiciels. Si vous souhaitez que AWS OpsWorks Stacks installe des packages ou effectue d'autres tâches de configuration au démarrage, vous devez fournir des recettes Chef personnalisées qui effectuent les tâches requises et les affecter aux événements de configuration des couches appropriées.
  - Instances basées sur Amazon EBS : AWS OpsWorks Stacks démarre une nouvelle instance avec la même AMI et rattache le volume racine, ce qui rétablit la configuration précédente de l'instance.
- Si vous annulez l'enregistrement d'une EC2 instance Amazon enregistrée, elle redevient une instance Amazon EC2 normale.

Instances locales

- AWS OpsWorks Stacks ne peut pas arrêter ou démarrer une instance locale enregistrée.

L'annulation de l'attribution d'une instance locale enregistrée déclenche un événement Shutdown. Cependant, cet événement exécute simplement les recettes Shutdown des couches affectées. Elles effectuent des tâches telles que l'arrêt des services, mais n'arrêtent pas l'instance.

- AWS OpsWorks Stacks ne peut pas réparer automatiquement une instance locale enregistrée en cas de défaillance, mais l'instance est marquée comme étant perdue.
- Les instances sur site ne peuvent pas utiliser les services d'adresse IP Elastic Load Balancing, Amazon EBS ou Elastic.

Attribution d'une instance enregistrée à une couche

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Note

Cette fonctionnalité est prise en charge uniquement pour les piles Linux.

Une fois que vous enregistrez une instance, vous pouvez l'affecter à une ou à plusieurs couches. L'avantage d'affecter une instance à une couche au lieu de la laisser non affectée est que vous pouvez attribuer des recettes personnalisées aux événements du cycle de vie de la couche. AWS OpsWorks Stacks les exécute ensuite automatiquement au moment opportun, après les recettes de la couche pour cet événement.

- Vous pouvez assigner n'importe quelle instance enregistrée à une couche personnalisée. Comme une couche personnalisée possède un ensemble minimal de recettes qui n'installent pas de packages, elle ne doit créer aucun conflit avec la configuration existante de l'instance.
- Vous pouvez attribuer des instances locales aux couches intégrées de AWS OpsWorks Stacks.

Chaque couche intégrée inclut des recettes qui installent automatiquement un ou plusieurs packages. Par exemple, les recettes de configuration de Java App Server installent Apache et Tomcat. Les recettes de la couche peuvent également effectuer d'autres opérations telles que le redémarrage de services et le déploiement d'applications. Avant d'attribuer une instance locale à une couche intégrée, vous devez vous assurer que les recettes de la couche ne créent aucun conflit, par exemple en essayant d'installer une version de serveur d'applications différente de celle qui se trouve actuellement sur l'instance. Pour plus d'informations, consultez Couches et AWS OpsWorks Référence de la couche Stacks.

Pour attribuer une instance enregistrée à une couche

1. Ajoutez à la pile les couches que vous souhaitez utiliser, si ce n'est déjà fait.
2. Choisissez Instances dans le volet de navigation, puis cliquez sur attribuer dans la colonne Actions de l'instance.
3. Sélectionnez les couches appropriées et choisissez Enregistrer.

Lorsque vous attribuez une instance à une couche, AWS OpsWorks Stacks effectue les opérations suivantes.

- Exécute les recettes Setup de la couche.
- Ajoute les adresses IP élastiques ou les volumes Amazon EBS attachés aux ressources de la pile.

  Vous pouvez ensuite utiliser AWS OpsWorks Stacks pour gérer ces ressources. Pour de plus amples informations, veuillez consulter Gestion des ressources.

Une fois qu'elles sont terminées, l'instance passe en ligne et est entièrement intégrée à la pile. AWS OpsWorks Stacks exécute ensuite les recettes attribuées à la couche chaque fois qu'un événement du cycle de vie se produit.

Annulation de l'enregistrement d'une instance

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

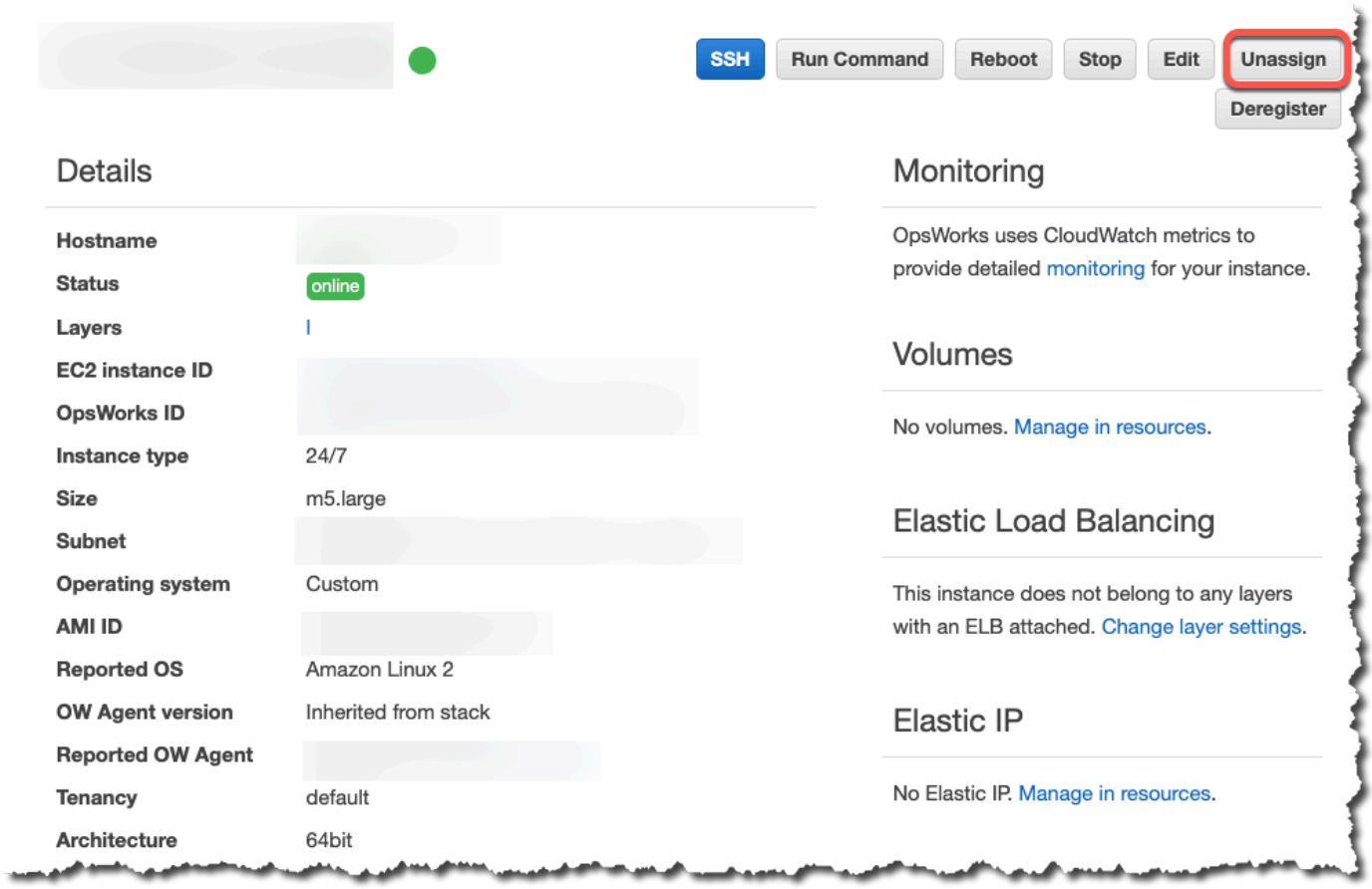
> **Note**
>
> Cette fonctionnalité est prise en charge uniquement pour les piles Linux.

Vous pouvez annuler l'attribution d'une instance enregistrée à ses couches à l'aide de la AWS OpsWorks console ou de AWS CLI l'opération du SDK.

Lorsque vous annulez l'attribution d'une instance, AWS OpsWorks Stacks exécute les recettes Shutdown de la couche sur l'instance. Ces recettes effectuent des tâches telles que l'arrêt des services, mais n'arrêtent pas l'instance. Si l'instance est attribuée à plusieurs couches, l'annulation s'applique à chaque couche ; vous ne pouvez pas annuler l'enregistrement d'une instance à partir d'un sous-ensemble de ses couches. Cependant, l'instance est toujours enregistrée auprès de la pile et vous pouvez l'affecter à une autre couche si vous le souhaitez.

Pour annuler l'attribution d'une instance enregistrée à l'aide de la console

1. Dans le panneau de navigation, choisissez Instances.
2. Choisissez l'instance dont vous souhaitez annuler l'attribution.
3. Sur la page Détails de l'instance, choisissez Annuler l'attribution.



Pour annuler l'attribution d'une instance enregistrée à l'aide du AWS CLI

Exécutez la aws opsworks unassign-instancecommande pour annuler l'attribution d'une instance enregistrée à toutes les couches qui l'utilisent.

```
aws opsworks unassign-instance --region region --instance-id instance-id
```

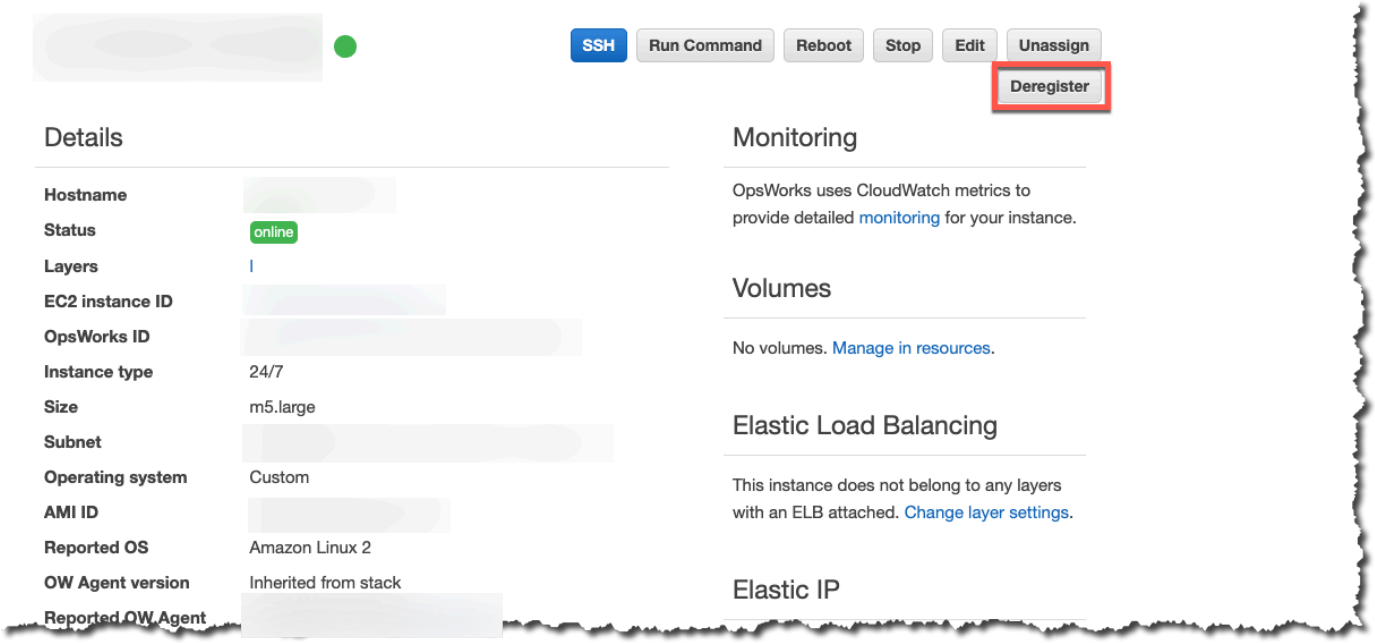
Annulation de l'enregistrement d'une instance

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez annuler l'enregistrement d'une instance à l'aide de la AWS OpsWorks console ou du SDK. AWS CLI

Pour désenregistrer une instance à l'aide de la console

1. Dans le panneau de navigation, choisissez Instances.
2. Choisissez l'instance que vous souhaitez désenregistrer.
3. Sur la page Détails de l'instance, choisissez Désenregistrer.



Pour annuler l'enregistrement d'une instance à l'aide du AWS CLI

Exécutez la aws opsworks deregister-instancecommande pour désenregistrer une instance de sa pile.

```
aws opsworks deregister-instance --region region --instance-id instance-id
```

Lorsque vous annulez l'enregistrement d'une instance, AWS OpsWorks Stacks effectue les opérations suivantes :

- Supprime l'instance de la pile.
- Annule l'enregistrement de l'instance à partir de couches assignées.
- Arrête et désinstalle l'agent.

- Désenregistre toutes les ressources associées (adresses IP élastiques et volumes Amazon EBS).

  Cette procédure inclut les ressources qui étaient attachées à l'instance avant l'enregistrement, ainsi que les ressources que vous avez utilisées AWS OpsWorks pour associer à l'instance alors qu'elle faisait partie de la pile. Après l'annulation de l'enregistrement, les ressources ne font plus partie des ressources de la pile, mais demeurent attachées à l'instance.
- Pour les instances sur site, arrête les frais.
- Supprime toutes les balises OpsWorks ajoutées à l'instance.

L'instance reste en cours d'exécution, mais elle est sous votre contrôle direct et n'est plus gérée par AWS OpsWorks Stacks.

> **Note**
>
> L'enregistrement et le désenregistrement d'ordinateurs ou d'instances ne sont entièrement pris en charge que dans les piles Linux. Pour les piles Windows, le désenregistrement des instances est autorisé, mais cela ne désinstalle pas l' OpsWorks agent de l'instance. L'annulation de l'enregistrement ne supprime pas tous les fichiers modifiés et ne restaure pas intégralement les copies sauvegardées de certains fichiers. Cette liste s'applique à la fois aux piles Chef 11.10 et Chef 12 ; les différences entre les deux versions sont mentionnées ici.
>
> - `/etc/hosts` est sauvegardé dans `/var/lib/aws/opsworks/local-mode-cache/backup/etc/`, mais n'est pas restauré.
> - Les entrées demeurent pour `aws` et `opsworks` dans les fichiers passwd, group et shadow, etc.
> - `/etc/sudoers`contient une référence à un répertoire AWS OpsWorks Stacks.
> - Les fichiers suivants peuvent être ignorés en toute sécurité ; à long terme, pensez à supprimer `/var/lib/aws/opsworks`.
>   - `/var/log/aws/opsworks` demeure sur les instances des piles Chef 11.10.
>   - `/var/lib/aws/opsworks` demeure sur les instances des piles Chef 11.10 et Chef 12.
>   - `/var/chef` demeure sur les instances des piles Chef 12.
> - Autres fichiers ignorés :
>   - `/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`

Cycle de vie des instances enregistrées

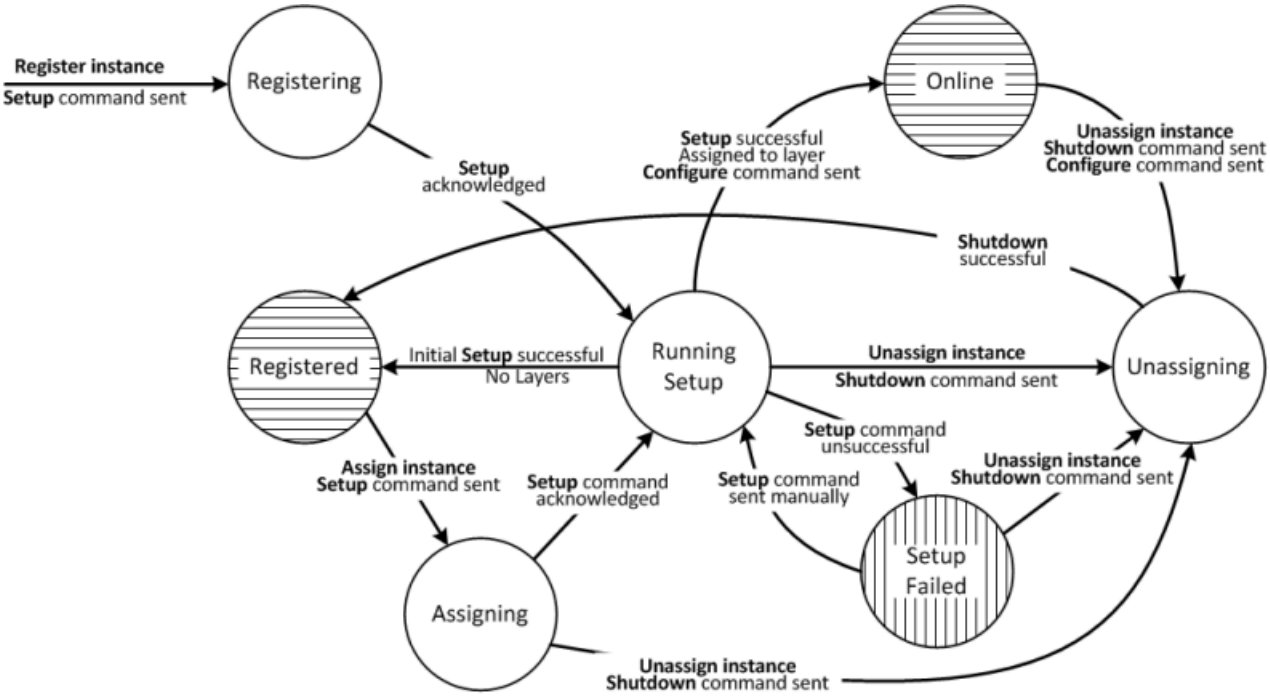
Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Note

Cette fonctionnalité est prise en charge uniquement pour les piles Linux.

Le cycle de vie d'une instance enregistrée commence une fois que l'agent est installé et en cours d'exécution. À ce stade, il demande à AWS OpsWorks Stacks d'enregistrer l'instance auprès de la pile. Le schéma d'état suivant résume les éléments clés du cycle de vie.



Chaque état correspond à un statut d'instance. Les arêtes représentent l'une des commandes AWS OpsWorks Stacks suivantes. Les détails sont détaillés dans les sections suivantes.

- Configuration : cette commande correspond à l'événement du cycle de vie de l'installation et exécute les recettes de configuration de l'instance.
- Configurer — Cette commande correspond à l'événement Configurer le cycle de vie.

  AWS OpsWorks Stacks déclenche cet événement sur chaque instance de la pile lorsqu'une instance entre ou quitte l'état en ligne. Les instances exécutent leurs recettes Configure, qui effectuent les modifications requises pour accueillir la nouvelle instance.
- Shutdown — Cette commande correspond à l'événement Shutdown Lifecycle, qui exécute les recettes d'arrêt de l'instance.

  Ces recettes effectuent des tâches telles que l'arrêt des services, mais elles n'arrêtent pas l'instance.
- Désenregistrer — Cette commande annule l'enregistrement d'une instance et ne correspond pas à un événement du cycle de vie.

> **Note**
>
> Pour des raisons de simplicité, le schéma n'affiche pas les états Deregistering et Deleted. Vous pouvez annuler l'enregistrement d'une instance à partir de l'un des états du diagramme, ce qui envoie une commande Deregister à l'instance et la passe à l'état Deregistering.
>
> - Si vous annulez l'enregistrement d'une instance en ligne, AWS OpsWorks Stacks envoie une commande Configure aux instances restantes de la pile pour les informer que l'instance est hors ligne.
> - Après la confirmation de la commande Deregister, l'instance continue à s'exécuter, mais elle est dans l'état Deleted et ne fait plus partie de la pile. Si vous souhaitez intégrer à nouveau l'instance dans la pile, vous devez la réenregistrer.

Rubriques

- Inscription en cours
- Running Setup
- Membre
- Assigning
- En ligne
- Setup Failed
- Unassigning
- Modifications de la configuration d'installation initiale

Inscription en cours

Une fois que l'agent a envoyé une demande d'enregistrement, AWS OpsWorks Stacks démarre le cycle de vie de l'instance en envoyant une commande de configuration à l'instance, en la plaçant dans l'état d'enregistrement. Une fois que l'instance reconnaît la commande Setup, elle passe à l'état Running Setup.

Running Setup

L'état Running Setup exécute les recettes Setup de l'instance. Le fonctionnement de Setup dépend de l'état précédent.

> **Note**
>
> Si vous annulez l'attribution de l'instance alors qu'elle est dans l'état Running Setup, AWS OpsWorks Stacks envoie une commande Shutdown, qui exécute les recettes d'arrêt de l'instance mais n'arrête pas l'instance. L'instance passe à l'état Unassigning.

Rubriques

- Inscription en cours
- Assigning
- Setup Failed

Inscription en cours

Au cours du processus d'enregistrement, le programme d'installation crée une instance AWS OpsWorks Stacks pour représenter l'instance enregistrée dans la pile et exécute un ensemble de recettes de configuration de base sur l'instance.

Un changement clé effectué par la configuration initiale est le remplacement du fichier hosts de l'instance. En enregistrant l'instance, vous avez transmis la gestion des utilisateurs à AWS OpsWorks Stacks, qui doit avoir son propre fichiers hôtes pour contrôler les autorisations de connexion SSH. La configuration initiale crée ou modifie également un certain nombre de fichiers et, sur les systèmes Ubuntu, modifie les sources des packages et installe un ensemble de packages. Pour plus de détails, consultez Modifications de la configuration d'installation initiale.

Lors de l'enregistrement, le processus appelle l'IAM `AttachUserPolicy` qui fait partie des autorisations associées à l'utilisateur IAM que vous créez comme condition préalable. Si `AttachUserPolicy` n'existe pas (probablement parce que vous exécutez une ancienne version de l'interface de ligne de commande AWS), le processus revient à l'appel `PutUserPolicy`.

> **Note**
>
> Pour des raisons de cohérence, AWS OpsWorks Stacks exécute toutes les recettes de configuration de base. Cependant, comme certaines d'entre elles n'effectuent tout ou partie de leurs tâches que si une instance a été affectée à une couche au moins, elles n'affectent pas nécessairement la configuration initiale.

- Si la commande Setup se déroule avec succès, l'instance passe à l'état Membre.
- Si la commande Setup échoue, l'instance passe à l'état Setup Failed.

Assigning

Au moins une couche est attribuée à l'instance. AWS OpsWorks Stacks exécute les recettes de configuration de chaque couche, y compris les recettes personnalisées que vous avez attribuées à l'événement de configuration des couches.

- Si la commande Setup se déroule avec succès, l'instance passe à l'état online (en ligne) et AWS OpsWorks Stacks déclenche un événement de cycle de vie Configure sur chaque instance de la pile afin de les informer de la nouvelle instance.
- Si la commande Setup échoue, l'instance passe à l'état Setup Failed.

> **Note**
>
> Ce processus d'installation exécute les recettes de base une deuxième fois. Toutefois, comme les recettes Chef sont idempotentes, elles ne répètent pas les tâches qui ont déjà été effectuées.

Setup Failed

Si le processus d'installation d'une instance dans l'état Assigning échoue, vous pouvez réessayer en utilisant la commande de pile Setup afin de relancer manuellement les recettes Setup de l'instance.

- Si la commande réussit, l'instance assignée passe à l'état En ligne et AWS OpsWorks Stacks déclenche un événement de cycle de vie Configure sur chaque instance de la pile pour les informer de la nouvelle instance.
- Si la commande échoue, l'instance revient à l'état Setup Failed.

Membre

Les instances à l'état Enregistré font partie de la pile et sont gérées par AWS OpsWorks Stacks mais ne sont attribuées à aucune couche. Elles peuvent rester indéfiniment dans cet état.

Si vous attribuez l'instance à une ou plusieurs couches, AWS OpsWorks Stacks envoie une commande de configuration à l'instance et elle passe à l'Assigningétat.

Assigning

Une fois que l'instance reconnaît la commande Setup, elle passe à l'état Running Setup.

Si vous annulez l'attribution de l'instance alors qu'elle est dans l'état Assignation, AWS OpsWorks Stacks met fin au processus de configuration et envoie une commande d'arrêt. L'instance passe à l'état Unassigning.

En ligne

L'instance est désormais membre d'au moins une couche et est traitée comme une instance AWS OpsWorks Stacks régulière. Elle peut demeurer indéfiniment dans cet état.

Si vous annulez l'attribution de l'instance alors qu'elle est en ligne, AWS OpsWorks Stacks envoie une commande d'arrêt à l'instance et une commande de configuration aux autres instances de la pile. L'instance passe à l'état Unassigning.

Setup Failed

La commande Setup a échoué.

- Vous pouvez réessayer en choisissant la commande de pile Setup.

  L'instance retourne à l'état Running Setup.
- Si vous annulez l'attribution de l'instance, AWS OpsWorks Stacks envoie une commande d'arrêt à l'instance.

  L'instance passe à l'état Unassigning.

Unassigning

Une fois que la commande Shutdown est terminée, l'instance n'est plus affectée à quelque couche que ce soit et retourne à l'état Membre.

> **Note**
>
> Si une instance est attribuée à plusieurs couches, l'annulation de l'attribution s'applique à chaque couche ; vous ne pouvez pas annuler l'attribution d'un sous-ensemble des couches assignées. Si vous souhaitez un autre ensemble de couches assignées, annulez l'attribution de l'instance, puis réassignez les couches souhaitées.

Modifications de la configuration d'installation initiale

L'installation initiale crée ou modifie les fichiers et répertoires suivants sur toutes les instances enregistrées.

Fichiers créés

```
/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/
```

Fichiers modifiés

```
/etc/apt/apt.conf.d/99-no-pipelining
/etc/crontab
/etc/default/monit
/etc/group
/etc/gshadow
/etc/monit/monitrc
/etc/passwd
/etc/security/limits.conf (removing limits only for EC2 micro instances)
/etc/shadow
/etc/sudoers
```

La configuration initiale crée également un fichier d'échange sur les EC2 micro-instances Amazon.

L'installation initiale apporte les modifications suivantes aux systèmes Ubuntu.

Sources de packages

L'installation initiale modifie les sources de packages comme suit :

- `deb http://archive.ubuntu.com/ubuntu/ ${code_name} main universe`

Pour: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name} main universe`

- `deb http://archive.ubuntu.com/ubuntu/ ${code_name}-updates main universe`

  Pour: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name}-updates main universe`
- `deb http://archive.ubuntu.com/ubuntu ${code_name}-security main universe`

  Pour: `deb-src http://archive.ubuntu.com/ubuntu ${code_name}-security main universe`
- `deb http://archive.ubuntu.com/ubuntu/ ${code_name}-updates multiverse`

  Pour: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name}-updates multiverse`
- `deb http://archive.ubuntu.com/ubuntu ${code_name}-security multiverse`

  Pour: `deb-src http://archive.ubuntu.com/ubuntu ${code_name}-security multiverse`
- `deb http://archive.ubuntu.com/ubuntu/ ${code_name} multiverse`

  Pour: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name} multiverse`
- `deb http://security.ubuntu.com/ubuntu ${code_name}-security multiverse`

  Pour: `deb-src http://security.ubuntu.com/ubuntu ${code_name}-security multiverse`

Packages

L'installation initiale désinstalle `landscape` et installe les packages suivants.

`autofs`	`libicu-dev`	`libopenssl-ruby`
`libssl-dev`	`libxml2-dev`	`libxslt-dev`
`libyaml-dev`	`monit`	`ntpd`

procps	ruby	ruby-dev
rubygems	screen	sqlite
vim	xfs	

Modification de la configuration de l'instance

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez modifier les configurations d'instance, y compris les instances Amazon Elastic Compute Cloud (Amazon EC2) enregistrées, dans les limites suivantes :

- L'instance doit être à l'état arrêté.

  Même si vous ne pouvez pas modifier les propriétés d'une instance en ligne, vous pouvez changer certains aspects de sa configuration en modifiant les couches de l'instance. Pour de plus amples informations, veuillez consulter Modification de OpsWorks la configuration d'une couche.
- Certains paramètres, tels que Availability Zone (Zone de disponibilité) et Scaling Type (Type de dimensionnement), sont déterminés lorsque vous créez l'instance et ne peuvent pas être modifiés ultérieurement.
- Certains paramètres peuvent être modifiés pour les instances basées sur le magasin uniquement, pas pour les instances basées sur Amazon Elastic Block Store.

  Par exemple, vous pouvez modifier le système d'exploitation d'une instance basée sur le stockage d'une instance. Les instances basées sur Amazon EBS doivent utiliser le système d'exploitation que vous avez spécifié lors de leur création. Pour plus d'informations sur le stockage d'instance, consultez Stockage.
- Par défaut, les instances héritent du paramètre de version de l'agent de la pile.

Vous pouvez utiliser la version de OpsWorks l'agent pour remplacer le paramètre de version de l'agent de la pile et spécifier une version d'agent particulière pour une instance. Si vous spécifiez la version de l'agent d'une instance, AWS OpsWorks Stacks ne met pas automatiquement à jour l'agent lorsqu'une nouvelle version est disponible, même si le paramètre de version de l'agent de la pile est Mise à jour automatique. Vous devez mettre à jour la version de l'agent de l'instance manuellement en modifiant la configuration de l'instance. AWS OpsWorks Stacks installe ensuite la version d'agent spécifiée sur l'instance.

> **Note**
>
> Vous ne pouvez pas modifier la configuration des instances locales enregistrées.

Pour modifier la configuration d'une instance

1. Arrêtez l'instance, si elle ne l'est pas déjà.
2. Sur la page Instances, cliquez sur un nom d'instance pour afficher la page Details (Détails).
3. Cliquez sur Edit (Modifier) pour afficher la page d'édition.
4. Modifiez la configuration de l'instance, le cas échéant.

Pour obtenir une description des paramètres Host name (Nom d'hôte), Size (Taille), SSH key (Clé SSH) et Operating system (Système d'exploitation), consultez Ajout d'une instance à une couche. Le paramètre Layers (Couches) vous permet d'ajouter ou de supprimer des couches. Les couches actuelles de l'instance apparaissent après la liste des couches.

- Pour ajouter une autre couche, sélectionnez-la dans la liste.
- Pour supprimer l'instance de l'une de ses couches, cliquez sur le x à côté de la couche appropriée.

  Comme une instance doit être membre d'au moins une couche, vous ne pouvez pas supprimer la dernière couche.

Lorsque vous redémarrez l'instance, AWS OpsWorks Stacks démarre une nouvelle EC2 instance Amazon avec la configuration mise à jour.

Supprimer des AWS OpsWorks instances de Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez utiliser AWS OpsWorks Stacks pour arrêter une instance, y compris les EC2 instances Amazon enregistrées. Cela arrête l' EC2 instance, mais celle-ci reste dans la pile. Vous pouvez la redémarrer en cliquant sur start (démarrer) dans la colonne Actions de l'instance. Si vous n'avez plus besoin d'une instance et que vous souhaitez la supprimer de la pile, vous pouvez la supprimer, ce qui supprime l'instance de la pile et met fin à l' EC2instance Amazon associée. La suppression d'une instance entraîne également la suppression de tous les journaux ou données associés, ainsi que de tous les volumes Amazon Elastic Block Store (EBS) présents sur l'instance.

> **Important**
>
> Cette rubrique s'applique uniquement aux EC2 instances Amazon gérées par AWS OpsWorks Stacks. Pour plus d'informations sur la façon de supprimer des instances gérées par la EC2 console ou l'API Amazon, consultez Terminate Your Instance.

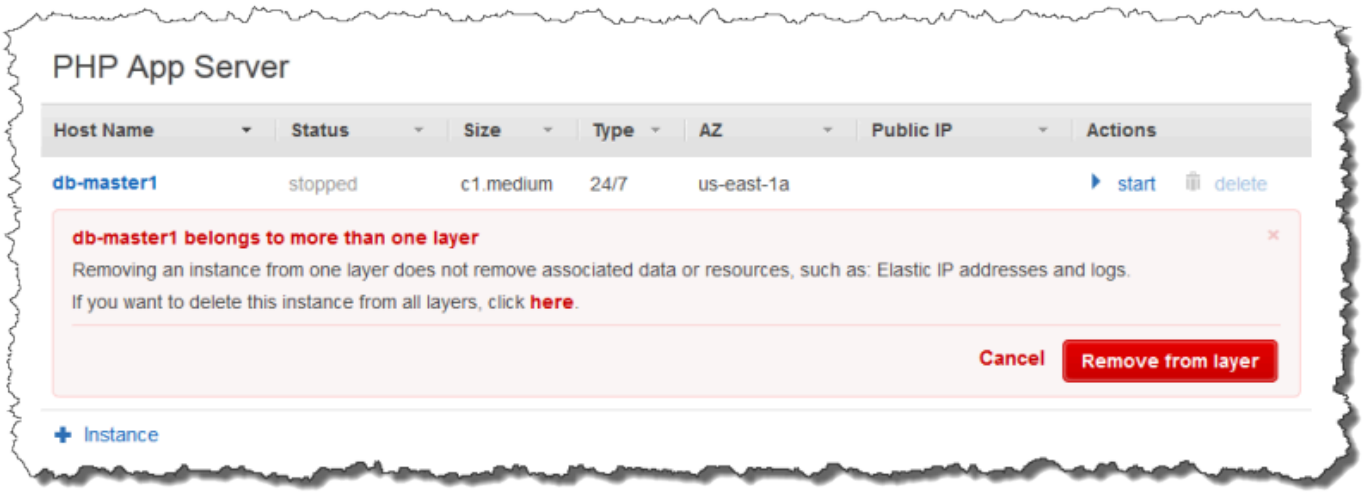
> **Note**
>
> Vous ne pouvez pas utiliser AWS OpsWorks Stacks pour supprimer une instance locale enregistrée.

Si une instance appartient à plusieurs niveaux, vous pouvez supprimer l'instance de la pile ou tout simplement supprimer une couche particulière. Vous pouvez également supprimer les couches d'une instance en modifiant sa configuration, comme décrit dans Modification de la configuration de l'instance.

> **Important**
>
> Vous devez supprimer les instances AWS OpsWorks Stacks uniquement à l'aide de la console ou de l' AWS OpsWorks API Stacks. En particulier, vous ne devez pas supprimer d'instances AWS OpsWorks Stacks à l'aide de la EC2 console ou de l'API Amazon, car les EC2 actions Amazon ne sont pas automatiquement synchronisées avec AWS OpsWorks Stacks. Par exemple, si la réparation automatique est activée et que vous mettez fin à une instance à l'aide de la EC2 console Amazon, AWS OpsWorks Stacks traite l'instance interrompue comme une instance défaillante et lance une autre EC2 instance Amazon pour la remplacer. Pour de plus amples informations, veuillez consulter Utilisation de la réparation automatique.

Pour supprimer une instance

1. Sur la page Instances, recherchez l'instance sous la couche appropriée. Si l'instance est en cours d'exécution, cliquez sur stop (arrêter) dans la colonne Actions.
2. Une fois que le statut devient stopped (arrêté), cliquez sur delete (supprimer). Si l'instance est membre de plusieurs couches, Layer AWS OpsWorks Stacks affiche la section suivante.



   - Pour supprimer l'instance de la seule couche sélectionnée, cliquez sur Remove from layer (Supprimer de la couche).

     L'instance demeure membre de ses autres couches et peut être redémarrée.
   - Pour supprimer l'instance de toutes ses couches, ce qui la supprime de la pile, cliquez ici.

3. Si vous choisissez de supprimer complètement une instance de la pile, ou si l'instance est membre d'une seule couche, AWS OpsWorks Stacks vous invite à confirmer la suppression.

   Choisissez Supprimer pour confirmer. Outre la suppression de l'instance de la pile, cette action supprime les journaux ou données associés, ainsi que les volumes racine attachés à l'instance. Pour supprimer tous les volumes d'instance, choisissez Supprimer les volumes EBS de l'instance (les instantanés ne seront pas supprimés) avant de choisir Supprimer.

Utilisation de SSH pour se connecter à une instance Linux

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez vous connecter à vos instances Linux en ligne via SSH à l'aide du MindTerm client intégré ou d'un client tiers, tel que PuTTY. SSH dépend généralement d'une paire de clés RSA pour l'authentification. Vous installez la clé publique sur l'instance et fournissez la clé privée correspondante au client SSH. AWS OpsWorks Stacks gère pour vous l'installation des clés publiques sur les instances de votre stack, comme suit.

- Paire de clés Amazon Elastic Compute Cloud (Amazon EC2) : si la région de la pile possède une ou plusieurs paires de EC2 clés Amazon, vous pouvez spécifier une paire de clés SSH par défaut pour la pile.

  Le cas échéant, vous pouvez remplacer la paire de clés par défaut et spécifier une paire différente lorsque vous créez une instance. Dans les deux cas, AWS OpsWorks Stacks installe la clé publique de la paire de clés spécifiée sur l'instance. Pour plus d'informations sur la création de paires de EC2 clés Amazon, consultez Amazon EC2 Key Pairs.
- Paire de clés personnelle — Chaque utilisateur peut enregistrer une paire de clés personnelle auprès de AWS OpsWorks Stacks.

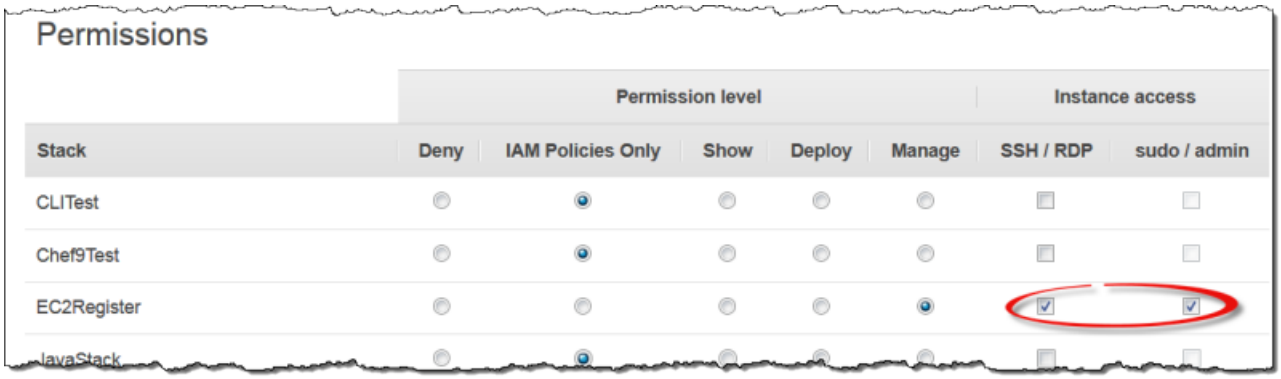
  L'utilisateur ou un administrateur enregistre la clé publique auprès de AWS OpsWorks Stacks, et l'utilisateur stocke la clé privée localement. Lors de la définition des autorisations pour une pile,

l'administrateur spécifie quels utilisateurs doivent avoir un accès SSH aux instances de la pile. AWS OpsWorks Stacks crée automatiquement un utilisateur système sur les instances de la pile pour chaque utilisateur autorisé et installe la clé publique personnelle de l'utilisateur.

Un utilisateur doit disposer d'une autorisation SSH pour utiliser le client MindTerm SSH ou pour utiliser sa paire de clés personnelles pour se connecter aux instances d'une pile.

Pour autoriser SSH pour un utilisateur

1. Dans le volet de navigation AWS OpsWorks Stacks, cliquez sur Autorisations.
2. Sélectionnez SSH/RDP pour l'utilisateur IAM souhaité afin d'accorder les autorisations nécessaires. Si vous souhaitez autoriser l'utilisateur à utiliser pour augmenter ses privilèges, par exemple **sudo** pour exécuter les commandes CLI de l'agent, sélectionnez également sudo/admin.



Pour plus d'informations sur l'utilisation de AWS OpsWorks Stacks pour gérer l'accès SSH, consultez. Gestion de l'accès SSH

Rubriques

- Utilisation du client MindTerm SSH intégré
- Utilisation d'un client SSH tiers

Utilisation du client MindTerm SSH intégré

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement

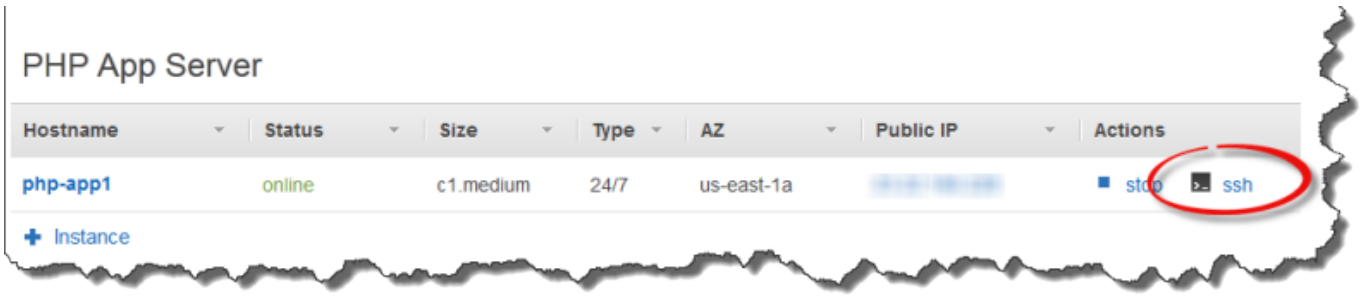
aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Le moyen le plus simple de se connecter à une instance Linux est d'utiliser le client MindTerm SSH intégré. Chaque instance en ligne inclut une action SSH que vous pouvez utiliser pour lancer le MindTerm client.

> **Note**
>
> Java doit être activé dans votre navigateur pour utiliser le MindTerm client.

Pour vous connecter avec le MindTerm client

1. Si vous ne l'avez pas déjà fait, autorisez l'accès SSH pour l'utilisateur IAM qui doit se connecter à l'instance, comme décrit dans la section précédente.
2. Connectez-vous en tant qu'utilisateur.
3. Sur la page Instances, choisissez ssh dans la colonne Actions de l'instance appropriée.



4. Pour la clé privée, indiquez le chemin d'accès à la clé privée personnelle de l'utilisateur ou à une clé EC2 privée Amazon, selon les clés publiques que vous avez installées sur l'instance.
5. Choisissez Lancer Mindterm et utilisez la fenêtre de terminal pour exécuter des commandes sur l'instance.

Utilisation d'un client SSH tiers

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez également utiliser un client SSH tiers, tel que PuTTY, pour vous connecter à une instance Linux.

Pour utiliser un client SSH tiers

1. Assurez-vous que AWS OpsWorks Stacks a installé une clé EC2 publique Amazon ou la clé publique personnelle d'un utilisateur IAM sur l'instance, comme indiqué précédemment.
2. Obtenez l'adresse IP publique ou le nom DNS public de l'instance sur la page des détails.
3. Fournissez au client le nom d'hôte de l'instance, qui dépend du système d'exploitation, comme suit :

   - Amazon Linux et Red Hat Enterprise Linux (RHEL) — `ec2-user@`*`DNSName/Address`*`.`
   - Ubuntu —`ubuntu@`*`DNSName/Address`*.

   *`DNSName/Address`*Remplacez-le par le nom DNS public ou l'adresse IP de l'étape précédente.
4. Fournissez au client une clé privée qui correspond à une clé publique installée. Vous pouvez utiliser une clé EC2 privée Amazon ou la clé privée personnelle d'un utilisateur IAM, selon les clés publiques installées sur l'instance.

Utilisation du protocole RDP pour se connecter à une instance Windows

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement

aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez utiliser le protocole RDP (Remote Desktop Protocol) Windows pour vous connecter à une instance Windows en ligne, comme suit :

- L'instance doit avoir un groupe de sécurité avec une règle entrante qui autorise l'accès RDP.

  Pour plus d'informations sur l'utilisation des groupes de sécurité , consultez Utilisation des groupes de sécurité.
- Utilisateurs ordinaires — AWS OpsWorks Stacks fournit aux utilisateurs ordinaires autorisés un mot de passe RDP valide pour une période limitée, qui peut aller de 30 minutes à 12 heures.

  En plus d'être autorisés, les utilisateurs doivent avoir au moins un niveau d'autorisation Afficher le niveau d'autorisation ou leurs politiques associées AWS Identity and Access Management (IAM) doivent autoriser l'`opsworks:GrantAccess`action.
- Administrateurs — Vous pouvez utiliser le mot de passe administrateur pour vous connecter pour une durée illimitée.

  Comme décrit plus loin, si vous avez spécifié une paire de clés Amazon Elastic Compute Cloud (Amazon EC2) pour l'instance, vous pouvez l'utiliser pour récupérer le mot de passe administrateur.

> **Note**
>
> Cette rubrique décrit comment utiliser le client Connexion Bureau à distance de Windows pour vous connecter à partir d'une station de travail Windows. Vous pouvez également utiliser l'un des clients RDP disponibles pour Linux ou OS X, mais la procédure peut être quelque peu différente. Pour plus d'informations sur les clients RDP compatibles avec Microsoft Windows Server 2012 R2, consultez Clients Bureau à distance Microsoft.

Rubriques

- Fourniture d'un groupe de sécurité qui permet l'accès RDP
- Connexion en tant qu'un utilisateur ordinaire

- Connexion en tant qu'Administrateur

Fourniture d'un groupe de sécurité qui permet l'accès RDP

Avant que vous puissiez utiliser RDP pour vous connecter à une instance Windows, les règles entrantes des groupes de sécurité de l'instance doivent autoriser les connexions RDP. Lorsque vous créez la première pile d'une région, AWS OpsWorks Stacks crée un ensemble de groupes de sécurité. Ils en incluent un nommé quelque chose comme`AWS-OpsWorks-RDP-Server`, que AWS OpsWorks Stacks attache à toutes les instances Windows pour permettre l'accès RDP. Cependant, comme, par défaut, ce groupe de sécurité n'a pas de règles, vous devez ajouter une règle entrante pour autoriser l'accès RDP à vos instances.

Pour autoriser l'accès RDP

1. Ouvrez la EC2console Amazon, définissez-la sur la région de la pile et choisissez Security Groups dans le volet de navigation.
2. Sélectionnez AWS- OpsWorks -RDP-Server, choisissez l'onglet Inbound, puis choisissez Edit.
3. Choisissez Add Rule (Ajouter une règle) et spécifiez les paramètres suivants :

   - Type — RDP
   - Source — Les adresses IP sources autorisées.

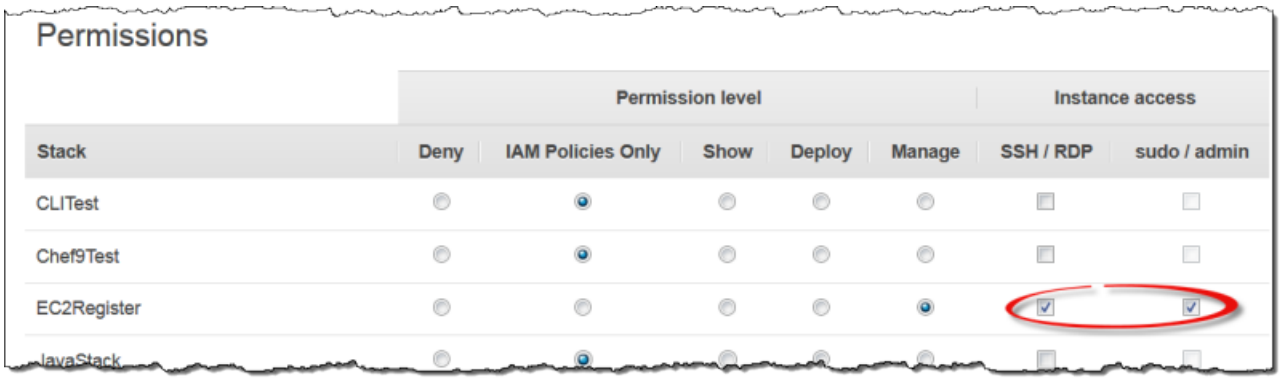
     Généralement, vous autorisez les demandes RDP entrantes de votre adresse IP ou plages d'adresses IP spécifiée (habituellement, votre plage d'adresses IP d'entreprise).

Connexion en tant qu'un utilisateur ordinaire

Un utilisateur autorisé peut se connecter aux instances à l'aide d'un mot de passe temporaire, fourni par AWS OpsWorks Stacks.

Pour autoriser le RDP pour un utilisateur ;

1. Dans le volet de navigation AWS OpsWorks Stacks, cliquez sur Autorisations.
2. Cochez la case SSH/RDP correspondant à l'utilisateur souhaité pour accorder les autorisations nécessaires. Si vous voulez que l'utilisateur dispose des autorisations d'administrateur, vous devez également sélectionner sudo/admin.



Les utilisateurs autorisés peuvent se connecter à l'une des instances en ligne de la pile, comme suit.

Pour vous connecter en tant qu'utilisateur IAM ordinaire

1. Connectez-vous en tant qu'utilisateur IAM.
2. Sur la page Instances, choisissez rdp dans la colonne Actions de l'instance appropriée.
3. Spécifiez la durée de la session, qui peut varier de 30 minutes à 12 heures, puis choisissez Generate Password (Générer le mot de passe). Le mot de passe est valide uniquement pour la durée de session spécifiée.
4. Enregistrez les valeurs nom DNS public, nom d'utilisateur et mot de passe, puis choisissez Accepter et fermer.
5. Ouvrez le client Connexion Bureau à distance Windows, choisissez Afficher les options et fournissez les informations suivantes à partir de celles que vous avez enregistrées à l'étape 4 :

   - Ordinateur : nom DNS public de l'instance.

     Vous pouvez aussi utiliser l'adresse IP publique, si vous préférez. Choisissez Instances et copiez l'adresse à partir de la colonne IP publique de l'instance.
   - Nom d'utilisateur : nom d'utilisateur.
6. Lorsque le client vous demande vos informations d'identification, entrez le mot de passe que vous avez enregistré à l'étape 4.

> **Note**
>
> AWS OpsWorks Stacks génère un mot de passe utilisateur uniquement pour les instances en ligne. Si vous démarrez une instance et que, par exemple, l'une de vos recettes Setup personnalisées échoue, l'instance se retrouve dans l'état `setup_failed`. Même si l'instance

n'est pas en ligne en ce qui concerne AWS OpsWorks Stacks, elle est en cours d'exécution et il est souvent utile de se connecter pour résoudre le problème. EC2 AWS OpsWorks Stacks ne générera pas de mot de passe pour vous dans ce cas, mais si vous avez attribué une paire de clés SSH à l'instance, vous pouvez utiliser la EC2 console ou la CLI pour récupérer le mot de passe administrateur de l'instance et vous connecter en tant qu'administrateur. Pour plus d'informations, consultez la section suivante.

Connexion en tant qu'Administrateur

Vous pouvez vous connecter à une instance en tant qu'Administrateur en utilisant le mot de passe approprié. Si vous avez attribué une paire de EC2 clés à une instance, Amazon l' EC2 utilise pour créer et chiffrer automatiquement un mot de passe administrateur au démarrage de l'instance. Vous pouvez ensuite utiliser la clé privée de la paire de clés avec la EC2 console, l'API ou la CLI pour récupérer et déchiffrer le mot de passe.

> **Note**
>
> Vous ne pouvez pas utiliser une paire de clés SSH personnelle pour récupérer un mot de passe administrateur ; vous devez utiliser une paire de EC2 clés.

Ce qui suit décrit comment utiliser la EC2 console pour récupérer un mot de passe administrateur et se connecter à une instance. Si vous préférez les outils de ligne de commande, vous pouvez également utiliser la commande `get-password-data` de l'interface de ligne de commande AWS pour récupérer le mot de passe.

Pour se connecter en tant qu'Administrateur

1. Assurez-vous d'avoir spécifié une paire de EC2 clés pour l'instance. Vous pouvez spécifier une paire de clés par défaut pour toutes les instances de la pile lorsque vous créez la pile, ou vous pouvez spécifier une paire de clés pour une instance particulière lorsque vous créez l'instance.
2. Ouvrez la EC2 console, définissez-la sur la région de la pile et choisissez Instances dans le volet de navigation.
3. Sélectionnez l'instance, choisissez Connexion, puis Obtenir le mot de passe.
4. Indiquez le chemin d'accès à la EC2 clé privée de la paire de clés sur votre poste de travail, puis choisissez Déchiffrer le mot de passe. Copiez le mot de passe déchiffré en vue d'une utilisation ultérieure.

5. Ouvrez le client Connexion Bureau à distance Windows, choisissez Afficher les options et fournissez les informations suivantes :
   - Ordinateur : nom DNS public ou adresse IP publique de l'instance, que vous pouvez obtenir sur la page de détails de l'instance.
   - Nom d'utilisateur —`Administrator`.
6. Lorsque le client vous demande vos informations d'identification, fournissez le mot de passe déchiffré de l'étape 4.

Applications

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une application AWS OpsWorks Stacks représente le code que vous souhaitez exécuter sur un serveur d'applications. Le code lui-même réside dans un référentiel tel qu'une archive Amazon S3 ; l'application contient les informations requises pour déployer le code sur les instances de serveur d'applications appropriées.

Lorsque vous déployez une application, AWS OpsWorks Stacks déclenche un événement Deploy, qui exécute les recettes de déploiement de chaque couche. AWS OpsWorks Stacks installe également des attributs de configuration et de déploiement de la pile qui contiennent toutes les informations nécessaires au déploiement de l'application, telles que le référentiel de l'application et les données de connexion à la base de données.

Vous devez implémenter des recettes personnalisées qui récupèrent les données de déploiement de l'application dans la configuration de la pile et les attributs de déploiement, et gèrent les tâches de déploiement.

Rubriques

- Ajout d'applications

- Déploiement d'applications
- Modification des applications
- Connexion d'une application à un serveur de base de données
- Utilisation des variables d'environnement
- Transmission de données aux applications
- Utilisation de clés SSH d'un référentiel Git
- Utilisation des domaines personnalisés
- Utilisation de SSL

Ajout d'applications

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

La première étape du déploiement d'une application sur vos serveurs d'applications consiste à ajouter une application à la pile. L'application contient une grande variété de métadonnées, comme le nom de l'application et le type, ainsi que les informations requises pour déployer l'application sur les instances serveur, telles que l'URL du référentiel. Vous devez gérer les autorisations Manage pour ajouter une application à une pile. Pour de plus amples informations, veuillez consulter Gestion des autorisations utilisateur.

> **Note**
>
> La procédure de cette section s'applique à Chef 12 et aux piles les plus récentes. Pour plus d'informations sur l'ajout d'applications aux couches des piles Chef 11, consultez Étape 2.4 : Créer et déployer une application - Chef 11.

Pour ajouter une application à une pile

1. Placez le code dans votre dépôt préféré : une archive Amazon S3, un dépôt Git, un dépôt Subversion ou une archive HTTP. Pour de plus amples informations, veuillez consulter Source de l'application.
2. Cliquez sur Apps (Applications) dans le panneau de navigation. Sur la page Apps (Applications), cliquez sur Add an app (Ajouter une application) pour votre première application. Pour les applications suivantes, cliquez sur +App.
3. Utilisez la page App New (Nouvelle application) pour configurer l'application, comme décrit dans la section suivante.

Configuration d'une application

La page Add App (Ajouter une application) se compose des sections suivantes : Settings (Paramètres), Application source (Source d'application), Data Sources (Sources de données), Environment Variables (Variables d'environnement), Add Domains (Ajouter des domaines) et SSL Settings (Paramètres SSL).

Rubriques

- Paramètres
- Source de l'application
- Sources de données
- Variables d'environnement
- Domaine et paramètres SSL

Paramètres

Nom

Le nom de l'application, qui est utilisé pour représenter l'application dans l'interface utilisateur. AWS OpsWorks Stacks utilise également ce nom pour générer un nom abrégé pour l'application utilisée en interne et pour identifier l'application dans la configuration de la pile et les attributs de déploiement. Une fois que vous avez ajouté l'application à la pile, vous pouvez voir le nom court en cliquant sur Apps (Applications) dans le panneau de navigation, puis sur le nom de l'application pour ouvrir la page des détails.

Document root (Racine du document)

AWS OpsWorks Stacks attribue le paramètre Document root à l'`[:document_root]`attribut dans les attributs de l'`deploy`application. La valeur par défaut est `null`. Vos recettes de déploiement peuvent obtenir cette valeur à partir des attributs `deploy` à l'aide de la syntaxe de nœud Chef standard et déployer le code spécifié à l'emplacement approprié sur le serveur. Pour plus d'informations sur la façon de déployer les applications, consultez Recettes Deploy.

Source de l'application

Vous pouvez déployer des applications à partir des types de référentiels suivants : Git, bundle Amazon S3, bundle HTTP et Other. Tous les types de référentiel nécessitent que vous spécifiiez le type du référentiel et l'URL du référentiel. Les types de référentiel individuels ont leurs propres exigences, comme expliqué ci-dessous.

> **Note**
>
> AWS OpsWorks Stacks déploie automatiquement les applications depuis les référentiels standard vers les couches de serveur intégrées. Si vous utilisez le type de référentiel Autre, qui est la seule option pour les piles Windows, AWS OpsWorks Stacks place les informations du référentiel dans les `deploy`attributs de l'application, mais vous devez implémenter des recettes personnalisées pour gérer les tâches de déploiement.

Rubriques

- Archive HTTP
- Archive Amazon S3
- Référentiel Git
- Autres référentiels

Archive HTTP

Pour utiliser un serveur HTTP accessible publiquement comme référentiel :

1. Créez une archive compressée (zip, gzip, bzip2, Java WAR ou tarball) du dossier contenant le code de l'application et tous les fichiers associés.

> **Note**
>
> AWS OpsWorks Stacks ne prend pas en charge les archives non compressées.

2. Chargez le fichier d'archive sur le serveur.
3. Pour spécifier le référentiel dans la console, sélectionnez HTTP Archive comme type de référentiel et saisissez l'URL.

   Si l'archive est protégée par mot de passe, sous Source de l'application, spécifiez les informations de connexion.

Archive Amazon S3

Pour utiliser un bucket Amazon Simple Storage Service comme référentiel :

1. Créez un compartiment Amazon S3 public ou privé. Pour plus d'informations, consultez la documentation Amazon S3.
2. Pour que AWS OpsWorks Stacks puisse accéder aux compartiments privés, vous devez être un utilisateur disposant au moins de droits en lecture seule sur le compartiment Amazon S3 et vous aurez besoin de l'ID de la clé d'accès et de la clé d'accès secrète. Pour plus d'informations, consultez la documentation AWS Identity and Access Management.
3. Placez le code et les fichiers associés dans un dossier, puis stockez le dossier dans une archive compressée (zip, gzip, bzip2, Java WAR ou tarball).

   > **Note**
   >
   > AWS OpsWorks Stacks ne prend pas en charge les archives non compressées.

4. Téléchargez le fichier d'archive dans le compartiment Amazon S3 et enregistrez l'URL.
5. Pour spécifier le référentiel dans la console AWS OpsWorks Stacks, définissez le type de référentiel sur S3 Archive et entrez l'URL de l'archive. Pour une archive privée, vous devez également fournir un ID de clé d'accès AWS et une clé d'accès secrète dont la stratégie accorde les autorisations pour accéder au compartiment. Laissez les paramètres en blanc pour les archives publiques.

Référentiel Git

Un dépôt Git permet le contrôle des sources et le contrôle des versions. AWS OpsWorks Stacks prend en charge les sites de dépôt hébergés publiquement tels que GitHubBitbucket, ainsi que les serveurs Git hébergés en privé. Pour les applications et les sous-modules Git, le format que vous utilisez pour spécifier l'URL du référentiel dans Application Source (Source d'application) dépend du fait que le référentiel est public ou privé :

Référentiel public : utilisez les protocoles en lecture seule HTTPS ou Git. Mise en route des piles Linux Chef 11Utilise, par exemple, un GitHub référentiel public accessible via l'un des formats d'URL suivants :

- Git en lecture seule : **`git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git`**
- HTTPS: **`https://github.com/amazonwebservices/opsworks-demo-php-simple-app.git`**

Référentiel privé : utilisez le format de lecture/écriture SSH illustré dans les exemples suivants :

- Référentiels GitHub : **`git@github.com:`*`project`*`/`*`repository`***.
- Référentiels sur un serveur Git : ***`user`*`@`*`server`*`:`*`project`*`/`*`repository`***

La sélection de Git sous Source Control (Contrôle de source) affiche deux paramètres facultatifs supplémentaires :

Repository SSH key (Clé SSH du référentiel)

Vous devez spécifier une clé SSH de déploiement pour accéder aux référentiels Git privés. Ce champ nécessite la clé privée ; la clé publique est attribuée à votre référentiel Git. Pour les sous-modules Git, la clé spécifiée doit avoir accès à ces sous-modules. Pour de plus amples informations, veuillez consulter Utilisation de clés SSH d'un référentiel Git.

> ⚠ Important
>
> La clé SSH de déploiement ne peut pas nécessiter de mot de passe ; AWS OpsWorks Stacks n'a aucun moyen de le transmettre.

Branch/Revision (Branche/Révision)

Si le dépôt comporte plusieurs branches, AWS OpsWorks Stacks télécharge la branche principale par défaut. Pour spécifier une branche en particulier, entrez le nom de la branche, le SHA1 hachage ou le nom de balise. Pour spécifier une validation particulière, saisissez l'identifiant de validation complet de 40 chiffres hexadécimaux.

Autres référentiels

Si les référentiels standard ne répondent pas à vos besoins, vous pouvez utiliser d'autres référentiels, comme Bazaar. Cependant, AWS OpsWorks Stacks ne déploie pas automatiquement les applications à partir de tels référentiels. Vous devez implémenter des recettes personnalisées pour gérer le processus de déploiement et les assigner aux événements Deploy des couches appropriées. Pour obtenir un exemple de la façon d'implémenter les recettes Deploy, consultez Recettes Deploy.

Sources de données

Cette section attache une base de données à l'application. Vous avez les options suivantes :

- RDS — Attachez l'une des couches de service Amazon RDS de la pile.
- Aucun : n'attachez pas de serveur de base de données.

Si vous sélectionnez RDS, vous devez spécifier les valeurs suivantes.

Instance de base de données

La liste inclut toutes les couches de service Amazon RDS. Vous pouvez aussi sélectionner l'une des valeurs suivantes :

(Obligatoire) Spécifiez le serveur de base de données à attacher à l'application. Le contenu de la liste dépend de la source de données.

- RDS — Liste des couches de service Amazon RDS de la pile.

Nom de base de données

(Facultatif) Spécifiez un nom de base de données.

- couche Amazon RDS : entrez le nom de base de données que vous avez spécifié pour l'instance Amazon RDS.

  Vous pouvez obtenir le nom de la base de données depuis la console Amazon RDS.

Lorsque vous déployez une application avec une base de données attachée, AWS OpsWorks Stacks ajoute la connexion de l'instance de base de données aux `deploy`attributs de l'application.

Vous pouvez écrire une recette personnalisée pour récupérer les informations des attributs `deploy` et les placer dans un fichier accessible par l'application. Il s'agit de la seule option pour fournir des informations de connexion de base de données au type d'application Other.

Pour plus d'informations sur la façon de gérer les connexions de base de données, consultez Connexion à une base de données.

Pour détacher un serveur de base de données d'une application, modifiez la configuration de l'application afin de spécifier un autre serveur de base de données ou aucun serveur.

Variables d'environnement

Vous pouvez spécifier un ensemble de variables d'environnement pour chaque application, spécifiques à l'application. Par exemple, si vous avez deux applications, les variables d'environnement que vous définissez pour la première application ne sont pas accessibles par la seconde application, et inversement. Vous pouvez également définir la même variable d'environnement pour plusieurs applications et lui attribuer une valeur différente pour chaque application.

> **Note**
>
> Il n'y a pas de limite spécifique au nombre de variables d'environnement. Toutefois, la taille de la structure de données associée, qui inclut les noms, les valeurs et les valeurs des indicateurs protégés des variables, ne peut pas dépasser 20 Ko. Cette limite doit convenir à la plupart des cas, si ce n'est tous. Son dépassement entraîne une erreur de service (console) ou une exception (API) avec le message, « Environment: is too large (maximum is 20KB) ».

AWS OpsWorks Stacks stocke les variables sous forme d'attributs dans les `deploy`attributs de l'application. Vous pouvez obtenir que vos recettes personnalisées récupèrent ces valeurs en utilisant la syntaxe de nœud Chef standard. Pour obtenir des exemples de la façon d'accéder aux variables d'environnement d'une application, consultez Utilisation des variables d'environnement .

Clé

Nom de la variable. Il peut contenir jusqu'à 64 caractères majuscules et minuscules, chiffres et traits de soulignement (_), mais il doit commencer par une lettre ou un trait de soulignement.

Valeur

Valeur de la variable. Elle peut contenir jusqu'à 256 caractères, qui doivent tous être affichables.

Valeur protégée

Indique si la valeur est protégée. Ce paramètre vous permet de masquer les informations sensibles telles que les mots de passe. Si vous définissez Protected value (Valeur protégée) pour une variable, une fois que vous avez créé l'application :

- La page des détails de l'application affiche uniquement le nom de la variable, pas la valeur.
- Si vous avez l'autorisation de modifier l'application, vous pouvez cliquer sur Update value (Mettre à jour la valeur) pour spécifier une nouvelle valeur, mais vous ne pouvez pas afficher ou modifier l'ancienne valeur.

> **Note**
>
> Les journaux de déploiement Chef peuvent parfois inclure des variables d'environnement. Cela signifie que des variables protégées peuvent s'afficher dans la console. Pour empêcher l'affichage de variables protégées dans la console, nous vous recommandons d'utiliser des compartiments Amazon S3 comme espace de stockage pour les variables protégées que vous ne souhaitez pas voir apparaître dans la console. Vous trouverez un exemple d'utilisation d'un compartiment S3 à cet effet dans la section Utilisation d'un compartiment Amazon S3 de ce guide.

Domaine et paramètres SSL

Pour le type d'application Autre, AWS OpsWorks Stacks ajoute les paramètres aux `deploy` attributs de l'application. Vos recettes peuvent récupérer les données à partir de ces attributs et configurer le serveur en fonction des besoins.

Paramètres de domaine

Cette section possède un champ facultatif Add Domains (Ajouter des domaines) pour spécifier les domaines. Pour de plus amples informations, veuillez consulter Utilisation des domaines personnalisés.

Paramètres SSL

Cette section possède un bouton bascule SSL Support (Prise en charge SSL) que vous pouvez utiliser pour activer ou désactiver SSL. Si vous cliquez sur Yes (Oui), vous devrez fournir les informations de certificat SSL. Pour de plus amples informations, veuillez consulter Utilisation de SSL.

Déploiement d'applications

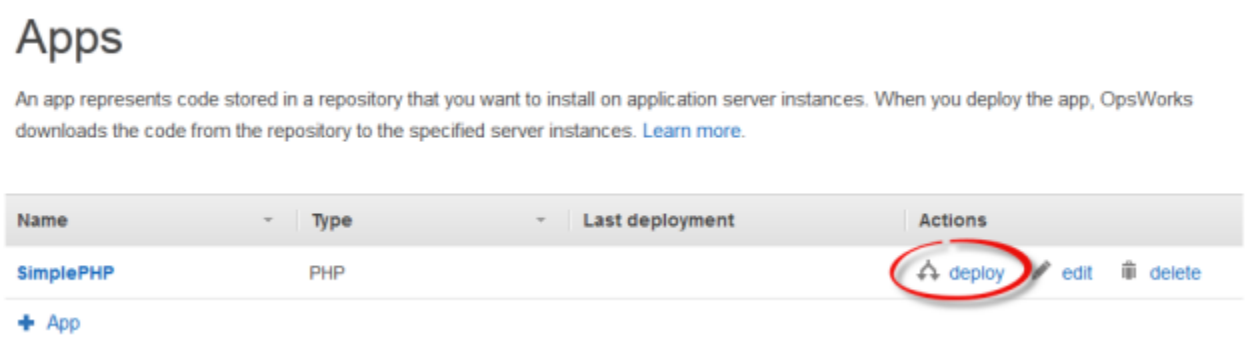
> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

L'objectif principal du déploiement consiste à déployer le code d'application et les fichiers associés aux instances de serveur d'application. L'opération de déploiement est gérée par les recettes Deploy de chaque instance, qui sont déterminées par la couche de l'instance.

Lorsque vous démarrez une instance, une fois les recettes de configuration terminées, AWS OpsWorks Stacks exécute automatiquement les recettes de déploiement de l'instance. Toutefois, lorsque vous ajoutez ou modifiez une application, vous devez la déployer manuellement sur toutes les instances en ligne. Vous devez disposer d'autorisations de gestion ou de déploiement pour déployer une application. Pour de plus amples informations, veuillez consulter Gestion des autorisations utilisateur.

Pour déployer une application

1. Sur la page Apps (Applications), cliquez sur l'action deploy (déployer) de l'application.



> **Note**
>
> Vous pouvez également déployer une application en cliquant sur Deployments (Déploiements) dans le panneau de navigation. Sur la page Deployments & Commands (Déploiements et commandes), cliquez sur Deploy an app (Déployer une application). Lors de cette opération, vous pouvez également choisir les applications à déployer.

2. Spécifiez les paramètres suivants :

   - (Obligatoire) Définissez Command: (Commande :) sur deploy (déployer), si cela n'a pas déjà été sélectionné.
   - (Facultatif) Incluez un commentaire.

3. Cliquez sur Avancé >> pour spécifier un JSON personnalisé. AWS OpsWorks Stacks ajoute un ensemble d'attributs de configuration et de déploiement de la pile à l'objet du nœud. Les attributs `deploy` contiennent les détails de déploiement et peuvent être utilisés par des recettes Deploy pour gérer l'installation et la configuration. Sur les piles Linux, vous pouvez utiliser le champ JSON personnalisé pour remplacer les paramètres par défaut des AWS OpsWorks piles ou transmettre des paramètres personnalisés à vos recettes personnalisées. Pour plus d'informations sur l'utilisation du JSON personnalisé, consultez Utilisation du JSON personnalisé.

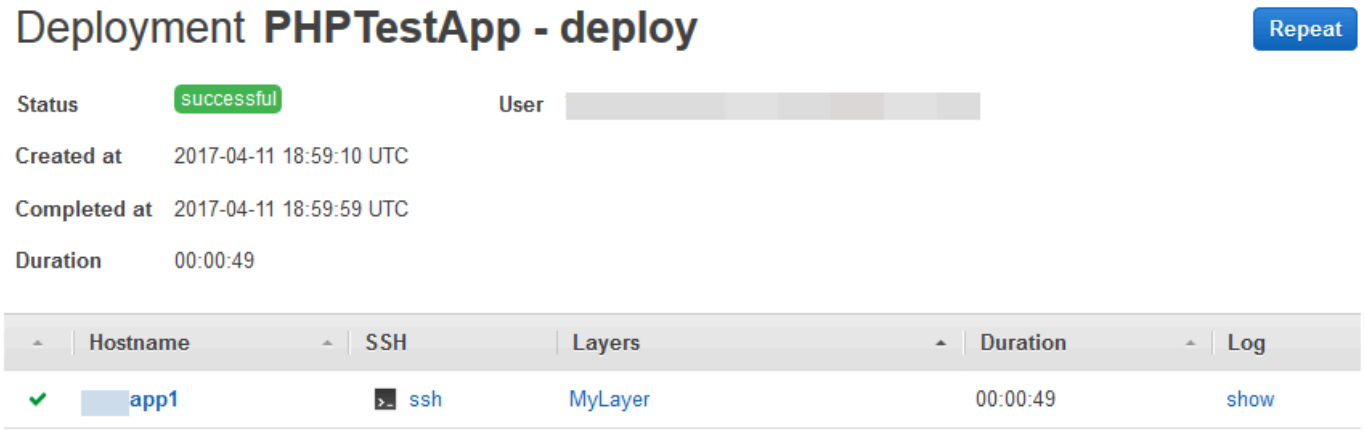
> **Note**
>
> Si vous spécifiez un JSON personnalisé ici, il est ajouté aux attributs de configuration et de déploiement de la pile pour ce déploiement uniquement. Si vous souhaitez ajouter un JSON personnalisé en permanence, vous devez l'ajouter à la pile. Le JSON personnalisé est limité à 120 Ko. Si vous avez besoin de plus de capacité, nous vous recommandons de stocker certaines données sur Amazon S3. Vos recettes personnalisées peuvent ensuite utiliser l'interface de ligne de commande AWS ou le kit SDK AWS pour Ruby

pour télécharger les données du compartiment vers votre instance. Pour obtenir un exemple, consultez Utilisation du kit SDK pour Ruby.

4. Sous Instances, cliquez sur Advanced (Avancé) >> et spécifiez les instances sur lesquelles exécuter la commande de déploiement.

   La commande de déploiement déclenche un événement Deploy qui exécute les déploiement des recettes sur les instances sélectionnées. Les recettes de déploiement pour le serveur d'application associé téléchargent le code et les fichiers liés depuis le référentiel et les installent sur l'instance, afin que vous sélectionniez toutes les instances de serveur d'application associées. Toutefois, d'autres types d'instance peuvent nécessiter des changements de configuration pour répondre aux besoins de la nouvelle application, c'est pourquoi il est souvent utile d'exécuter les recettes de déploiement sur ces instances aussi. Ces recettes mettent à jour la configuration en fonction des besoins, mais n'installent pas les fichiers de l'application. Pour plus d'informations sur les recettes, consultez Livres de recettes et recettes.

5. Cliquez sur Deploy (Déployer) pour exécuter les recettes de déploiement sur les instances spécifiés, ce qui affiche la page de déploiement. Lorsque le processus est terminé, AWS OpsWorks Stacks marque l'application d'une coche verte pour indiquer un déploiement réussi. Si le déploiement échoue, AWS OpsWorks Stacks marque l'application d'un X rouge. Dans ce cas, vous pouvez accéder à la page Déploiements et consulter le journal de déploiement pour plus d'informations.



> **Note**
>
> Lorsque vous déployez une mise à jour sur une application JSP, il est possible que Tomcat ne reconnaisse pas la mise à jour et continue à exécuter la version d'application existante. Cela peut se produire, par exemple, si vous déployez votre application en tant que fichier .zip

qui contient uniquement une page JSP. Afin de garantir que Tomcat exécute la version la plus récemment déployée, le répertoire racine du projet doit inclure un répertoire WEB-INF qui contient un fichier `web.xml`. Un fichier `web.xml` peut contenir une grande variété de contenus, mais ce qui suit est suffisant pour veiller à ce que Tomcat reconnaisse les mises à jour et exécute la version d'application actuellement déployée. Vous n'avez pas à changer la version pour chaque mise à jour. Tomcat reconnaît la mise à jour même si la version n'a pas changé.

```
<context-param>
  <param-name>appVersion</param-name>
  <param-value>0.1</param-value>
</context-param>
```

Autres commandes de déploiement

La page Deploy app (Déployer une application) inclut plusieurs autres commandes pour la gestion de vos applications et des serveurs associés. Parmi les commandes suivantes, seule la commande Undeploy (Annuler un déploiement) est disponible pour les applications sur des piles Chef 12.

Annulation du déploiement

Déclenche un événement du cycle de vie Undeploy, qui exécute les recettes d'annulation de déploiement pour supprimer toutes les versions de l'application des instances spécifiées.

Restauration

Restaure la version d'application précédemment déployée. Par exemple, si vous avez déployé l'application trois fois et que vous exécutez ensuite Rollback (Restaurer), le serveur sert l'application du deuxième déploiement. Si vous exécutez Rollback (Restaurer) à nouveau, le serveur sert l'application du premier déploiement. Par défaut, AWS OpsWorks Stacks stocke les cinq déploiements les plus récents, ce qui vous permet de revenir en arrière jusqu'à quatre versions. Si vous dépassez le nombre de versions stockées, la commande échoue et laisse la version la plus ancienne en place. Cette commande n'est pas disponible dans des piles Chef 12.

Start Web Server

Exécute les recettes qui démarrent le serveur d'application sur les instances spécifiées. Cette commande n'est pas disponible dans des piles Chef 12.

Stop Web Server

Exécute les recettes qui arrêtent le serveur d'application sur les instances spécifiées. Cette commande n'est pas disponible dans des piles Chef 12.

Restart Web Server

Exécute les recettes qui redémarrent le serveur d'application sur les instances spécifiées. Cette commande n'est pas disponible dans des piles Chef 12.

Important

Start Web Server (Démarrer le serveur web), Stop Web Server (Arrêter le serveur web), Restart Web Server (Redémarrer le serveur web) et Rollback (Restaurer) sont globalement des versions personnalisées de la commande de pile Execute Recipes (Exécuter les recettes). Elles exécutent un ensemble de recettes qui effectuent la tâche sur les instances spécifiées.

- Ces commandes ne déclenchent pas d'événement du cycle de vie, vous ne pouvez donc pas les connecter pour exécuter du code personnalisé.
- Ces commandes fonctionnent uniquement pour les couches de serveur d'application intégrées.

  En particulier, ces commandes n'ont aucune incidence sur les couches personnalisées, même si elles prennent en charge un serveur d'application. Pour démarrer, arrêter ou redémarrer des serveurs sur une couche personnalisée, vous devez implémenter des recettes personnalisées pour réaliser ces tâches et utiliser la commande de pile Execute Recipes (Exécuter les recettes) pour les exécuter. Pour plus d'informations sur l'implémentation et l'installation des recettes personnalisées, consultez Livres de recettes et recettes.

Modification des applications

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement

aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez modifier la configuration d'une application en modifiant l'application. Par exemple, si vous êtes prêt à déployer une nouvelle version, vous pouvez modifier les paramètres AWS OpsWorks Stacks de l'application pour utiliser la nouvelle branche du référentiel. Vous devez avoir les autorisations Manage ou Deploy pour modifier la configuration d'une application. Pour de plus amples informations, veuillez consulter Gestion des autorisations utilisateur.

Pour modifier une application

1. Sur la page Apps (Applications), cliquez sur le nom de l'application pour ouvrir la page des détails.
2. Cliquez sur Edit (Modifier) pour modifier la configuration de l'application.

   - Si vous modifiez le nom de l'application, AWS OpsWorks Stacks utilise le nouveau nom pour identifier l'application dans la console.

     La modification du nom ne change pas le nom court associé. Le nom court est défini lorsque vous ajoutez l'application à la pile et ne peut pas être modifié par la suite.
   - Si vous avez spécifié une variable d'environnement protégée, vous ne pouvez pas afficher ou modifier la valeur. Cependant, vous pouvez spécifier une nouvelle valeur en cliquant sur Update value (Mettre à jour la valeur).
3. Cliquez sur Save (Enregistrer) pour enregistrer la nouvelle configuration, puis sur Deploy App (Déployer l'application) pour déployer l'application.

La modification d'une application modifie les paramètres de AWS OpsWorks Stacks, mais n'affecte pas les instances de la pile. Lorsque vous déployez une application pour la première fois, les recettes Deploy téléchargent le code et les fichiers apparentés sur les instances du serveur d'applications, qui exécutent alors la copie locale. Si vous modifiez l'application dans le référentiel ou si vous modifiez d'autres paramètres, vous devez déployer l'application pour installer les mises à jour sur vos instances de serveur d'applications, comme suit. AWS OpsWorks Stacks déploie automatiquement la version actuelle de l'application sur les nouvelles instances lors de leur démarrage. Pour les instances existantes, cependant, la situation est différente :

- AWS OpsWorks Stacks déploie automatiquement la version actuelle de l'application sur les nouvelles instances lors de leur démarrage.
- AWS OpsWorks Stacks déploie automatiquement la dernière version de l'application sur les instances hors ligne, y compris les instances basées sur la charge et le temps, lors de leur redémarrage.
- Vous devez déployer manuellement l'application mise à jour sur les instances en ligne.

Pour plus d'informations sur la façon de déployer des applications, consultez Déploiement d'applications

Connexion d'une application à un serveur de base de données

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez associer un serveur de base de données Amazon RDS à une application lorsque vous créez l'application ou ultérieurement en la modifiant. Votre application peut ensuite utiliser les informations de connexion à la base de données : nom d'utilisateur, mot de passe,... : pour se connecter au serveur de base de données. Lorsque vous déployez une application, AWS OpsWorks Stacks fournit ces informations aux applications de deux manières :

- Pour les piles Linux, AWS OpsWorks Stacks crée un fichier sur chacune des instances du serveur d'applications intégré contenant les données de connexion que l'application peut utiliser pour se connecter au serveur de base de données.
- AWS OpsWorks Stacks inclut les informations de connexion dans la configuration de la pile et les attributs de déploiement installés sur chaque instance.

  Vous pouvez implémenter une recette personnalisée pour extraire les informations de connexion de ces attributs et les placer dans un fichier selon votre format préféré. Pour de plus amples informations, veuillez consulter Transmission de données aux applications.

> **Important**
>
> Pour les piles Linux, si vous souhaitez associer une couche de service Amazon RDS à votre application, vous devez ajouter le package de pilotes approprié à la couche de serveur d'applications associée, comme suit :
>
> 1. Cliquez sur Layers (Couches) dans le panneau de navigation et ouvrez l'onglet Recipes (Recettes) du serveur d'applications.
> 2. Cliquez sur Edit (Modifier) et ajoutez le package de pilotes approprié aux OS Packages (Packages de SE). Par exemple, vous devez spécifier `mysql` si la couche contient des instances Amazon Linux et `mysql-client` si la couche contient des instances Ubuntu.
> 3. Enregistrez les modifications et redéployez l'application.

Utilisation d'une recette personnalisée

Vous pouvez implémenter une recette personnalisée qui extrait les données de connexion des attributs `deploy` de l'application et les enregistre sous une forme que l'application peut lire, telle qu'un fichier YAML, par exemple.

Vous attachez un serveur de base de données à une application lorsque vous créez l'application ou ultérieurement en modifiant l'application. Lorsque vous déployez l'application, AWS OpsWorks Stacks installe une configuration de pile et des attributs de déploiement sur chaque instance qui incluent les informations de connexion à la base de données. Votre application peut ensuite récupérer les attributs appropriés. Les détails varient selon que vous utilisez une pile Linux ou Windows.

Connexion à un serveur de base de données pour une pile Linux

Pour les piles Linux, l'espace de `deploy` noms des attributs de configuration et de déploiement de la pile inclut un attribut pour chaque application déployée, nommé avec le nom abrégé de l'application. Lorsque vous attachez un serveur de base de données à une application, AWS OpsWorks Stacks renseigne l'`[:database]`attribut de l'application avec les informations de connexion et les installe sur les instances de la pile pour chaque déploiement ultérieur. Les valeurs d'attribut sont fournies par l'utilisateur ou générées par AWS OpsWorks Stacks.

> **Note**
>
> AWS OpsWorks Stacks vous permet d'associer un serveur de base de données à plusieurs applications, mais chaque application ne peut être associée qu'à un seul serveur de base

de données. Si vous souhaitez connecter une application à plusieurs serveurs de base de données, attachez l'un des serveurs à l'application et utilisez les informations des attributs `deploy` de l'application pour vous connecter au serveur. Utilisez le JSON personnalisé pour transmettre les informations de connexion des autres serveurs de base de données à l'application. Pour de plus amples informations, veuillez consulter Transmission de données aux applications.

Une application peut utiliser les informations de connexion des attributs `deploy` de l'instance pour se connecter à une base de données. Toutefois, les applications ne peuvent pas accéder directement à ces informations : seules les recettes peuvent accéder aux attributs. `deploy` Vous pouvez résoudre ce problème en mettant en œuvre une recette personnalisée qui extrait les informations de connexion à partir des attributs `deploy` et les place dans un fichier qui peut être lu par l'application.

Utilisation des variables d'environnement

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Note

Les recommandations de cette rubrique s'appliquent à Chef 11.10 et versions antérieures. Pour obtenir les variables d'environnement dans Chef 12 et versions plus récentes, vous devez utiliser le conteneur de données de l'application. Pour plus d'informations, consultez AWS OpsWorks Data Bag Reference et App Data Bag (aws_opsworks_app).

Lorsque vous spécifiez des variables d'environnement pour une application, AWS OpsWorks Stacks ajoute les définitions des variables aux `deploy`attributs de l'application.

Les couches personnalisées peuvent utiliser une recette pour récupérer une valeur de variable grâce à une syntaxe nœud standard, et la stocker sous une forme accessible par les applications de la couche.

Vous devez implémenter une recette personnalisée qui obtient les valeurs des variables d'environnement à partir des attributs `deploy` de l'instance. La recette peut alors stocker les données sur l'instance sous une forme accessible par l'application, telle qu'un fichier YAML. Les définitions des variables environnement d'une application sont stockées dans les attributs `deploy`, dans les variables `environment_variables` de l'environnement. L'exemple suivant montre l'emplacement de ces attributs pour une application nommée `simplephpapp`, à l'aide de JSON pour représenter la structure des attributs.

```
{
  ...
  "ssh_users": {
  },
  "deploy": {
    "simplephpapp": {
      "application": "simplephpapp",
      "application_type": "php",
      "environment_variables": {
        "USER_ID": "168424",
        "USER_KEY": "somepassword"
      },
    ...
  }
}
```

Une recette peut obtenir les valeurs variables en utilisant un syntaxe nœud standard. L'exemple suivant montre comment obtenir la valeur `USER_ID` depuis le JSON précédent et le placer dans le journal Chef.

```
Chef::Log.info("USER_ID: #{node[:deploy]['simplephpapp'][:environment_variables]
[:USER_ID]}")
```

Pour une description plus détaillée de la façon de récupérer les informations depuis le JSON de configuration et de déploiement de pile, et le stocker sur l'instance, consultez Transmission de données aux applications.

Transmission de données aux applications

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Il est souvent utile de transmettre des données telles que les paires clé-valeur à une application sur le serveur. Pour ce faire, utilisez JSON personnalisé pour ajouter les données à la pile. AWS OpsWorks Stacks ajoute les données à l'objet nœud de chaque instance pour chaque événement du cycle de vie.

Cependant, notez que, même si les recettes peuvent obtenir le JSON personnalisé à partir de l'objet de nœud à l'aide des attributs Chef, les applications ne le peuvent pas. Une approche pour obtenir les données JSON personnalisées d'une ou de plusieurs applications consiste à mettre en place une recette personnalisée qui extrait les données à partir de l'objet node et les écrit dans un fichier que l'application peut lire. L'exemple dans cette rubrique montre comment écrire les données dans un fichier YAML, mais vous pouvez utiliser la même approche de base pour d'autres formats, tels que JSON ou XML.

Pour passer les données clé-valeur aux instances de la pile, ajoutez le JSON personnalisé comme suit à la pile. Pour plus d'informations sur l'ajout du JSON personnalisé à une pile, consultez Utilisation du JSON personnalisé.

```
{
  "my_app_data": {
    "app1": {
      "key1": "value1",
      "key2": "value2",
      "key3": "value3"
    },
    "app2": {
      "key1": "value1",
      "key2": "value2",
```

```
      "key3": "value3"
    }
  }
}
```

L'exemple suppose que vous ayez deux applications dont les noms courts sont `app1` et `app2`, chacune ayant trois valeurs de données. La recette d'accompagnement présume que vous utilisiez les noms courts des applications pour identifier les données associées ; les autres noms sont arbitraires. Pour plus d'informations sur les noms courts d'application, consultez Paramètres.

La recette de l'exemple suivant montre comment extraire les données de chaque application à partir des attributs `deploy` et les placer dans un fichier `.yml`. La recette part du principe que votre JSON personnalisé contient les données de chaque application.

```
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
```

Les attributs `deploy` contiennent un attribut pour chaque application, nommé d'après le nom court de l'application. Chaque attribut d'application contient un ensemble d'attributs qui représentent différentes informations sur l'application. Cet exemple utilise le répertoire de déploiement de l'application, représenté par l'attribut `[:deploy][:app_short_name][:deploy_to]`. Pour plus d'informations sur `[:deploy]`, consultez Attributs deploy.

Pour chaque application de `deploy`, la recette effectue les opérations suivantes :

1. Crée un fichier nommé `app_data.yml` dans le sous-répertoire `shared/config` du répertoire `[:deploy_to]` de l'application.

   Pour plus d'informations sur la façon dont AWS OpsWorks Stacks installe les applications, consultez. Recettes Deploy
2. Convertit les valeurs JSON personnalisées de l'application en YAML et écrit les données mises en forme sur `app_data.yml`.

Pour transmettre les données à une application

1. Ajoutez une application à la pile et notez son nom court. Pour de plus amples informations, veuillez consulter Ajout d'applications.
2. Ajoutez le JSON personnalisé avec les données de l'application aux attributs `deploy`, comme indiqué plus tôt. Pour plus d'informations sur l'ajout du JSON personnalisé à une pile, consultez Utilisation du JSON personnalisé.
3. Créez un livre de recettes et ajoutez une recette avec le code basé sur l'exemple précédent, modifié en fonction des besoins pour les noms d'attributs que vous avez utilisés dans le JSON personnalisé. Pour plus d'informations sur la création de livres de recettes et de recettes, consultez Livres de recettes et recettes. Si vous avez déjà des livres personnalisés pour cette pile, vous pouvez aussi ajouter la recette à un livre de recettes existant, ou même ajouter le code à une recette Deploy existante.
4. Installez le livre de recettes sur votre pile. Pour de plus amples informations, veuillez consulter Installation de livres de recettes personnalisés.
5. Attribuez la recette à l'événement Deploy Lifecycle de la couche du serveur d'applications. AWS OpsWorks Stacks exécutera ensuite la recette sur chaque nouvelle instance, après son démarrage. Pour de plus amples informations, veuillez consulter Exécution des recettes.
6. Déployez l'application, qui installe également les attributs de configuration et de déploiement de pile, lesquels contiennent désormais vos données.

> **Note**
>
> Si les fichiers de données doivent être en place avant le déploiement de l'application, vous pouvez également assigner la recette à l'événement du cycle Setup de la couche, qui se produit une fois, juste après la fin du démarrage de l'instance. Cependant, AWS OpsWorks Stacks n'aura pas encore créé les répertoires de déploiement. Votre recette doit donc créer les répertoires requis de manière explicite avant de créer le fichier de données. L'exemple suivant crée explicitement le répertoire `/shared/config` de l'application, puis crée un fichier de données dans ce répertoire.

```
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
```

Pour charger les données, vous pouvez utiliser, par exemple, le code Sinatra suivant :

```
#!/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
```

Vous pouvez mettre à jour les valeurs des données de l'application à tout moment en modifiant le JSON personnalisé, comme suit.

Pour mettre à jour les données de l'application

1. Modifiez le JSON personnalisé pour mettre à jour les valeurs des données.
2. Déployez à nouveau l'application, qui indique à AWS OpsWorks Stacks d'exécuter les recettes de déploiement sur les instances de la pile. Comme les recettes utilisent les attributs extraits des attributs de configuration et de déploiement mis à jour de la pile, votre recette personnalisée met à jour les fichiers de données avec les valeurs actuelles.

Utilisation de clés SSH d'un référentiel Git

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une clé SSH de référentiel Git, parfois appelée clé SSH de déploiement, est une clé SSH sans mot de passe qui permet d'accéder à un référentiel Git privé. Idéalement, elle n'appartient à aucun développeur spécifique. Son objectif est de permettre à AWS OpsWorks Stacks de déployer de manière asynchrone des applications ou des livres de recettes à partir d'un dépôt Git sans avoir besoin d'informations supplémentaires de votre part.

La section suivante décrit la procédure de base pour créer une clé SSH de référentiel. Pour plus d'informations, consultez la documentation de votre référentiel. Par exemple, la section Gestion des clés de déploiement décrit comment créer une clé SSH de référentiel pour un GitHub référentiel, et les clés de déploiement sur Bitbucket décrit comment créer une clé SSH de référentiel pour un référentiel Bitbucket. Notez qu'une partie de la documentation décrit la création d'une clé sur un serveur. Pour AWS OpsWorks Stacks, remplacez simplement « serveur » par « poste de travail » dans les instructions.

Pour créer une clé SSH de référentiel

1. Créez une paire de clés SSH de déploiement pour votre référentiel Git de votre station de travail à l'aide d'un programme tel que `ssh-keygen`.

   > **Important**
   >
   > AWS OpsWorks Stacks ne prend pas en charge les phrases de passe clés SSH.

2. Attribuez la clé publique au référentiel et stockez la clé privée sur votre station de travail.
3. Entrez la clé privée dans la zone Repository SSH Key (Clé SSH du référentiel) lorsque vous ajoutez une application ou que vous spécifiez le référentiel du livre de recettes. Pour de plus amples informations, veuillez consulter Ajout d'applications.

AWS OpsWorks Stacks transmet la clé SSH du référentiel à chaque instance, et les recettes intégrées utilisent ensuite la clé pour se connecter au référentiel et télécharger le code. La clé est stockée dans les attributs `deploy` comme `node[:deploy]['appshortname'][:scm][:ssh_key]` et n'est accessible que par l'utilisateur racine.

Utilisation des domaines personnalisés

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Si vous hébergez un nom de domaine avec un tiers, vous pouvez mapper ce nom de domaine à une application. La procédure de base est la suivante :

1. Créez un sous-domaine avec votre serveur d'inscriptions DNS et mappez-le à l'adresse IP Elastic de votre équilibreur de charge ou à l'adresse IP publique de votre serveur d'applications.
2. Mettez à jour la configuration de votre application pour qu'elle pointe vers le sous-domaine et redéployez l'application.

> **Note**
>
> Assurez-vous que vous transférez votre nom de domaine non qualifié (par exemple, myapp1.example.com) vers votre nom de domaine complet (par exemple, www.myapp1.example.com) afin que les deux soient mappés à votre application.

Lorsque vous configurez un domaine pour une application, il est répertorié comme alias serveur dans le fichier de configuration du serveur. Si vous utilisez un équilibreur de charge, celui-ci vérifie le nom de domaine de l'URL lorsque les demandes lui parviennent et redirige le trafic en fonction du domaine.

Pour mapper un sous-domaine à une adresse IP

1. Si vous utilisez un équilibreur de charge, sur la page Instances, cliquez sur l'instance de l'équilibreur de charge pour ouvrir la page des détails et obtenir l'adresse IP Elastic de l'instance. Sinon, obtenez l'adresse IP publique à partir de la page des détails de l'instance serveur d'applications.
2. Suivez les instructions fournies par votre serveur d'inscriptions DNS pour créer et mapper votre sous-domaine à l'adresse IP de l'étape 1.

> **Note**
>
> Si l'instance de l'équilibreur de charge se termine à un moment donné, une nouvelle adresse IP Elastic vous est attribuée. Vous devez mettre à jour vos paramètres de serveur d'inscriptions DNS pour mapper avec la nouvelle adresse IP Elastic.

AWS OpsWorks Stacks ajoute simplement les paramètres du domaine aux `deploy`attributs de l'application. Vous devez implémenter une recette personnalisée pour extraire les informations de l'objet nœud et configurer le serveur correctement. Pour de plus amples informations, veuillez consulter Livres de recettes et recettes.

Exécution de plusieurs applications sur le même serveur d'applications

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Les informations contenues dans cette rubrique ne s'appliquent pas aux applications Node.js.

Si vous avez plusieurs applications du même type, il est parfois plus rentable de les exécuter sur les mêmes instances de serveur d'applications.

Pour exécuter plusieurs applications sur le même serveur

1. Ajoutez une application à la pile pour chaque application.
2. Obtenez un sous-domaine distinct pour chaque application et mappez les sous-domaines à l'adresse IP du serveur d'applications ou de l'équilibreur de charge.
3. Modifiez la configuration de chaque application pour spécifier le sous-domaine approprié.

Pour plus d'informations sur l'exécution de ces tâches, consultez Utilisation des domaines personnalisés.

> **Note**
>
> Si vos serveurs d'applications exécutent plusieurs applications HTTP, vous pouvez utiliser Elastic Load Balancing pour l'équilibrage de charge. Dans le cas de plusieurs applications HTTPS, vous devez mettre fin à la connexion SSL sur l'équilibreur de charge ou créer une pile distincte pour chaque application. Les requêtes HTTPS sont chiffrées, ce qui signifie que si vous mettez fin à la connexion SSL sur les serveurs, l'équilibreur de charge ne peut pas vérifier le nom de domaine pour déterminer quelle application doit gérer la requête.

Utilisation de SSL

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Pour utiliser SSL avec votre application, vous devez obtenir un certificat de serveur numérique auprès d'une autorité de certification. Pour simplifier, cette procédure crée un certificat, qu'elle signe elle-même. Les certificats signés automatiquement sont utiles à des fins d'apprentissage et de test,

mais vous devez toujours utiliser un certificat signé par une autorité de certification pour les piles en production.

Dans cette procédure, vous allez exécuter les actions suivantes :

1. Installer et configurer OpenSSL.
2. Créer une clé privée.
3. Créer une demande de signature de certificat.
4. Générer un certificat signé automatiquement.
5. Modifier l'application avec les informations de votre certificat.

> **Important**
>
> Si votre application utilise le protocole SSL, nous vous recommandons de le désactiver SSLv3, si possible, dans les couches de votre serveur d'applications afin de corriger les vulnérabilités décrites dans le document CVE-2014-3566. Si votre stack inclut une couche Ganglia, vous devez également désactiver SSL v3 pour cette couche. Les détails dépendent de la couche particulière ; pour plus d'informations, consultez les éléments suivants.
>
> - Java App Server AWS OpsWorks Stacks Layer
> - Node.js App Server AWS OpsWorks Stacks Layer
> - couche d' AWS OpsWorks empilements de serveurs d'applications PHP
> - Rails App Server AWS OpsWorks Stacks Layer
> - Couche Static Web Server AWS OpsWorks Stacks
> - Couche ganglionnaire

Rubriques

- Étape 1 : Installer et configurer OpenSSL.
- Étape 2 : Créer une clé privée
- Étape 3 : Créer une demande de signature de certificat
- Étape 4 : Envoyer la demande de signature de certificat à une autorité de certification
- Étape 5 : Modifier l'application

Étape 1 : Installer et configurer OpenSSL.

La création et le chargement de certificats de serveur nécessitent un outil qui prend en charge les protocoles SSL et TLS. OpenSSL est un outil open source qui fournit les fonctions cryptographiques de base requises pour créer un jeton RSA et vous connecter avec votre clé privée.

La procédure suivante suppose que votre ordinateur n'ait pas déjà OpenSSL installé.

Pour installer OpenSSL sous Linux et Unix

1. Accédez à OpenSSL : source, tarballs.
2. Téléchargez la source la plus récente.
3. Créez le package.

Pour installer OpenSSL sous Windows

1. Si le package redistribuable Microsoft Visual C++ 2008 n'est pas déjà installé sur votre système, téléchargez-le.
2. Exécutez le programme d'installation et suivez les instructions fournies par l'Assistant d'installation Microsoft Visual C++ 2008 Redistributable pour installer le redistribuable.
3. Accédez à OpenSSL : distributions binaires, cliquez sur la version appropriée des fichiers binaires OpenSSL pour votre environnement et enregistrez le programme d'installation localement.
4. Exécutez le programme d'installation et suivez les instructions de l'assistant de configuration d'OpenSSL pour installer les fichiers binaires.

Créez une variable d'environnement qui pointe sur le point d'installation OpenSSL en ouvrant une fenêtre de terminal ou de commande, puis en utilisant les lignes de commande suivantes.

- Sous Linux et Unix

```
export OpenSSL_HOME=path_to_your_OpenSSL_installation
```

- Sous Windows

```
set OpenSSL_HOME=path_to_your_OpenSSL_installation
```

Ajoutez le chemin d'accès des fichiers binaires OpenSSL à la variable path de votre ordinateur en ouvrant une fenêtre de terminal ou de commande, puis en utilisant les lignes de commande suivantes.

- Sous Linux et Unix

```
export PATH=$PATH:$OpenSSL_HOME/bin
```

- Sous Windows

```
set Path=OpenSSL_HOME\bin;%Path%
```

> **Note**
>
> Les modifications apportées aux variables d'environnement à l'aide de ces lignes de commande sont valides uniquement pour la session de ligne de commande actuelle.

Étape 2 : Créer une clé privée

Vous avez besoin d'une clé privée unique pour créer votre demande de signature de certificat (CSR). Créez la clé à l'aide de la ligne de commande suivante :

```
openssl genrsa 2048 > privatekey.pem
```

Étape 3 : Créer une demande de signature de certificat

Un fichier de demande de signature de certificat (CSR) est un fichier envoyé à une autorité de certification pour demander un certificat de serveur numérique. Créez le fichier CSR à l'aide de la ligne de commande suivante.

```
openssl req -new -key privatekey.pem -out csr.pem
```

La sortie de la commande doit se présenter comme suit :

```
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.
```

Le tableau suivant peur vous aider à créer votre propre demande de certificat.

Données de la demande de certificat

Name (Nom)	Description	Exemple
Nom du pays	Abréviation ISO de deux lettres de votre pays.	US = États-Unis
État ou Province	Nom de l'état ou de la province où votre organisation se situe. Ce nom ne peut pas être abrégé.	Washington
Nom de la localité	Nom de la ville où votre organisation se situe.	Seattle
Nom de l'organisation	Nom légal complet de votre organisation. N'abrégez pas le nom de votre organisation.	CorporationX
Unité d'organisation	(Facultatif) Pour des informations supplémentaires sur l'organisation.	Marketing
Nom commun	Nom de domaine complet de votre CNAME. Vous recevrez un avertissement de vérification du nom du certificat si la correspondance n'est pas exacte.	www.exemple.com
Adresse e-mail	Adresse électronique de l'administrateur du serveur.	quelquun@exemple.com

> **Note**
>
> Le champ Nom commun est souvent mal compris et rempli de manière incorrecte. Le nom commun est généralement votre hôte plus le nom du domaine. Il se présente comme « www.exemple.com » ou « exemple.com ». Vous devez créer une demande CSR qui utilise votre nom commun correct.

Étape 4 : Envoyer la demande de signature de certificat à une autorité de certification

Pour une utilisation en production, vous obtenez un certificat de serveur en soumettant votre demande CSR à une autorité de certification, ce qui peut nécessiter d'autres informations d'identification ou preuves d'identité. En cas de réussite, l'autorité de certification renvoie un certificat d'identité signé numériquement et, éventuellement, un fichier de chaîne de certificat. AWS ne recommande pas de CA spécifique. Pour une liste partielle des fournisseurs disponibles CAs, voir Autorité de certification - Fournisseurs sur Wikipedia.

Vous pouvez également générer un certificat signé automatiquement, qui peut être utilisé à des fins de test uniquement. Pour cet exemple, utilisez la ligne de commande suivante pour générer un certificat signé automatiquement.

```
openssl x509 -req -days 365 -in csr.pem -signkey privatekey.pem -out server.crt
```

La sortie sera similaire à l'exemple suivant :

```
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
```

Étape 5 : Modifier l'application

Une fois que vous avez généré et signé votre certificat, mettez à jour votre application pour activer le protocole SSL et fournissez les informations sur votre certificat. Sur la page Apps (Applications), choisissez une application pour ouvrir la page des détails, puis cliquez sur Edit App (Modifier l'application). Pour activer la prise en charge SSL, définissez Enable SSL (Activer SSL) sur Yes (Oui). Les options de configuration suivantes s'affichent.

SSL Certificate (Certificat SSL)

Collez le contenu du fichier de certificat de clé publique (.crt) dans la zone. Le certificat doit ressembler à ce qui suit :

```
-----BEGIN CERTIFICATE-----
MIICuTCCAiICCQCtqFKItVQJpzANBgkqhkiG9w0BAQUFADCBoDELMAkGA1UEBhMC
dXMxEzARBgNVBAgMCndhc2hpbmd0b24xEDAOBgNVBAcMB3NlYXR0bGUxDzANBgNV
BAoMBmFtYXpvbjEWMBQGA1UECwwNRGV2IGFuZCBUb29sczEdMBsGA1UEAwwUc3Rl
cGhhbmllYXBpZXJjZS5jb20xIjAgBgkqhkiG9w0BCQEWE3NhcGllcmNlQGFtYXpv
...
-----END CERTIFICATE-----
```

> **Note**
>
> Si vous utilisez Nginx et que vous avez un fichier de chaîne de certificat, vous devez ajouter le contenu au fichier de certificat de clé publique.

Si vous mettez à jour un certificat existant, procédez comme suit :

- Choisissez Update SSL certificate (Mettre à jour le certificat SSL) pour mettre à jour le certificat.
- Si le nouveau certificat ne correspond pas à la clé privée existante, choisissez Update SSL Certificate Key (Mettre à jour la clé de certificat SSL).
- Si le nouveau certificat ne correspond pas à la chaîne de certificat existante, choisissez Update SSL certificates (Mettre à jour les certificats SSL).

SSL Certificate Key (Clé de certificat SSL)

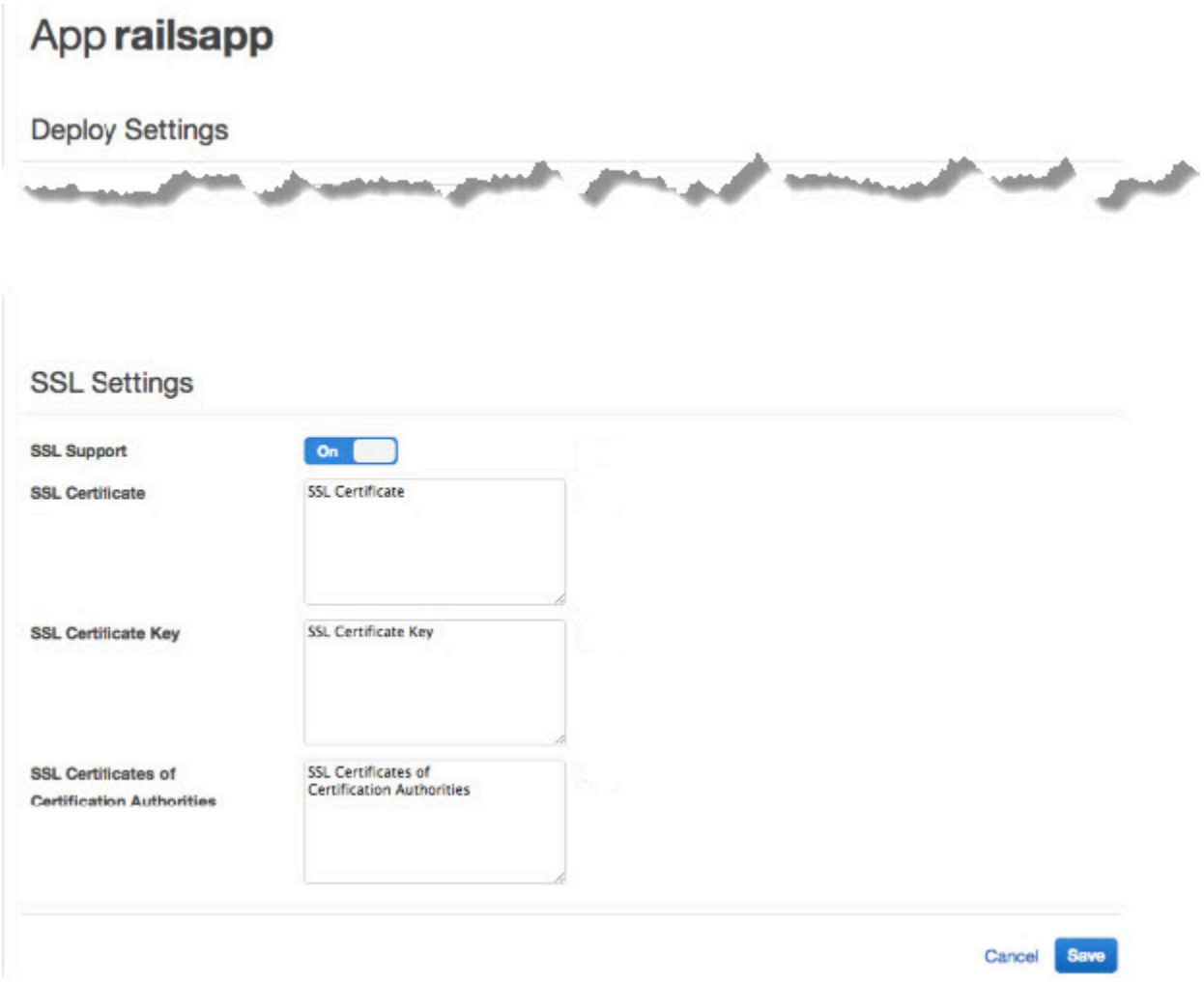
Collez le contenu du fichier de clé privé (fichier .pem) dans la zone. Elles doivent se présenter comme suit :

```
----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQC0CYklJY5r4vV2NHQYEpwtsLuMMBhylMrgBShKq+HHVLYQQCL6
+wGIiRq5qXqZlRXje3GM5Jvcm6q0R71MfRIl1FuzKyqDtneZaAIEYniZibHiUnmO
/UNqpFDosw/6hY3ONk0fSBlU4ivD0Gjpf6J80jL3DJ4R23Ed0sdL4pRT3QIDAQAB
AoGBAKmMfWrNRqYVtGKgnWB6Tji9QrKQLMXjmHeGg95mppdJELiXHhpMvrHtpIyK
...
-----END RSA PRIVATE KEY-----
```

SSL certificates of Certification Authorities (Certificats SSL des autorités de certification)

Si vous avez un fichier de chaîne de certificats, collez le contenu dans la zone.

> **Note**
>
> Si vous utilisez Nginx, vous devez laisser cette zone vide. Si vous avez un fichier de chaîne de certificats, ajoutez-le au fichier de certificat de clé publique dans SSL Certificate (Certificat SSL).



Après avoir cliqué sur Save (Enregistrer), redéployez l'application pour mettre à jour vos instances en ligne.

Pour les couches de serveur d'applications intégrées, AWS OpsWorks Stacks met automatiquement à jour la configuration du serveur. Une fois que le déploiement est terminé, vous pouvez vérifier que votre installation OpenSSL a fonctionné, comme suit.

Pour vérifier une installation OpenSSL

1. Accédez à la page Instances.
2. Exécutez l'application en cliquant sur l'adresse IP de l'instance du serveur d'applications ou, si vous utilisez un équilibreur de charge, son adresse IP.
3. Modifiez le préfixe de l'adresse IP de **`http://`** en **`https://`** et actualisez le navigateur pour vérifier que la page se charge correctement avec SSL.

Les utilisateurs qui ont configuré des applications pour qu'elles s'exécutent dans Mozilla Firefox obtiennent parfois le certificat d'erreur suivant : `SEC_ERROR_UNKNOWN_ISSUER`. Cette erreur peut être provoquée par une fonctionnalité de remplacement de certificat dans les programmes antivirus et anti-logiciels malveillants de votre entreprise, par certains types de logiciels de surveillance et de filtrage du trafic réseau, ou par des logiciels malveillants. Pour plus d'informations sur la façon de résoudre cette erreur, consultez Que faire devant l'erreur « SEC_ERROR_UNKNOWN_ISSUER » sur des sites web sécurisés sur le site web du support de Mozilla Firefox.

Pour toutes les autres couches, y compris les couches personnalisées, AWS OpsWorks Stacks ajoute simplement les paramètres SSL aux attributs `deploy` de l'application. Vous devez implémenter une recette personnalisée pour extraire les informations de l'objet nœud et configurer le serveur correctement.

Livres de recettes et recettes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks utilise les livres de recettes Chef pour gérer des tâches telles que l'installation et la configuration de packages et le déploiement d'applications. Cette section décrit comment utiliser les livres de recettes avec AWS OpsWorks Stacks. Pour plus d'informations, consultez Chef.

> **Note**
>
> AWS OpsWorks Stacks prend actuellement en charge les versions 12, 11.10.4, 11.4.4 et 0.9.15.5 de Chef. Toutefois, Chef 0.9.15.5 est obsolète et nous vous déconseillons de l'utiliser pour de nouvelles piles. Pour plus de commodité, elles sont généralement désignées par leurs numéros de version majeure et de version mineure. Les piles exécutant Chef 0.9 ou 11.4 utilisent Chef Solo et les piles exécutant Chef 12 ou 11.10 utilisent Chef Client en mode local. Pour les piles Linux, vous pouvez utiliser le Gestionnaire de configuration et spécifier la version Chef à utiliser quand vous créez une pile. Les piles Windows doivent utiliser Chef 12.2. Pour plus d'informations, y compris les instructions sur la migration des piles vers des versions Chef plus récentes, consultez Versions Chef.

Rubriques

- Référentiels de livres de recettes
- Versions Chef
- Versions de Ruby
- Installation de livres de recettes personnalisés
- Mise à jour des livres de recettes personnalisés
- Exécution des recettes

Référentiels de livres de recettes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vos livres de recettes personnalisés doivent être stockés dans un référentiel en ligne, soit une archive telle qu'un fichier .zip, soit un gestionnaire de contrôle de source tel que Git. Une pile peut avoir un seul référentiel de livres personnalisés, mais ce référentiel peut contenir n'importe quel nombre de livres de recettes. Lorsque vous installez ou mettez à jour les livres de recettes, AWS OpsWorks Stacks installe l'intégralité du référentiel dans un cache local sur chacune des instances de la pile. Par exemple, lorsqu'une instance a besoin d'exécuter une ou plusieurs recettes, elle utilise le code à partir du cache local.

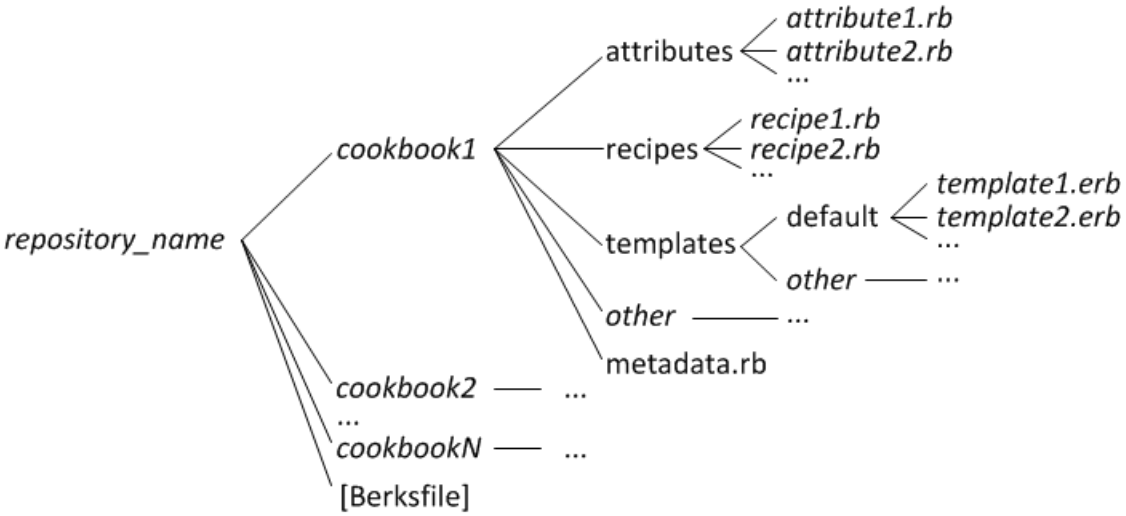
Ce qui suit décrit la structure de votre référentiel de livres de recettes, ce qui dépend du type. Le texte en italique dans les illustrations représente les noms de répertoires et de fichiers définis par l'utilisateur, y compris le nom du référentiel ou de l'archive.

Gestionnaire de contrôle de source

AWS OpsWorks Stacks prend en charge les gestionnaires de contrôle de source suivants :

- Stacks Linux — Git et Subversion
- piles Windows — Git

L'exemple suivant montre la structure de répertoires et de fichiers requise :



- Les répertoires de livres de recettes doivent être au niveau supérieur.

Archivage

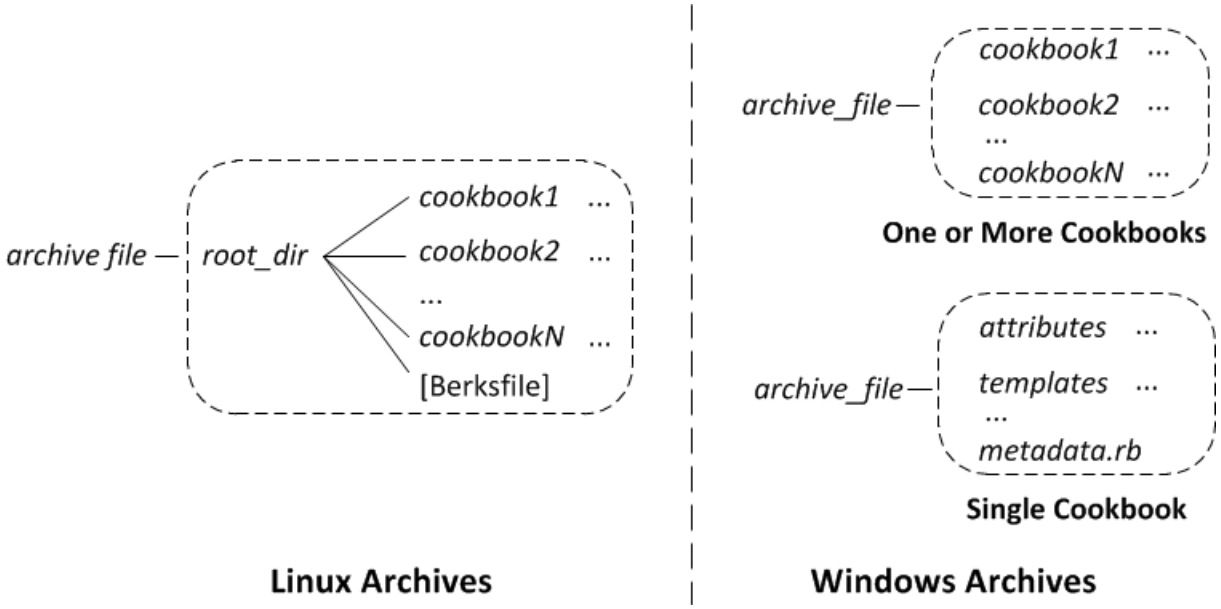
AWS OpsWorks Stacks prend en charge les archives suivantes :

- Stacks Linux : fichiers zip, gzip, bzip2 ou tarball stockés sur Amazon S3 ou sur un site Web (archive HTTP).

  AWS OpsWorks Stacks ne prend pas en charge les archives non compressées.

- Windows Stacks : fichiers zip et tgz (tar compressé au format gzip), stockés sur Amazon S3.

L'exemple suivant montre le répertoire et la structure de fichiers requis, en fonction de l'exécution d'une pile Linux ou Windows. La structure du livre de recettes est la même que pour les référentiels SCM, c'est pourquoi elle est représentée par une ellipse (...).



- Linux Stacks — Les répertoires des livres de recettes doivent être contenus dans un répertoire racine.
- Windows Stacks — Les livres de recettes doivent se trouver au niveau supérieur de l'archive.

  Si vous avez un seul livre de recettes, vous pouvez ignorer de façon facultative le répertoire des livres de recettes et mettre les fichiers du livre au niveau supérieur. Dans ce cas, AWS OpsWorks Stacks obtient le nom du livre de recettes à partir de metadata.rb.

Chaque répertoire de livre de recettes a au moins et généralement tous les fichiers et répertoires standard suivants, qui doivent utiliser des noms standard :

- `attributes`— Les fichiers d'attributs du livre de recettes.
- `recipes`— Les fichiers de recettes du livre de recettes.
- `templates`— Les fichiers modèles du livre de recettes.
- *`other`*— Répertoires facultatifs définis par l'utilisateur qui contiennent d'autres types de fichiers, tels que des définitions ou des spécifications.
- `metadata.rb`— Les métadonnées du livre de recettes.

Pour Chef 11.10 et les versions ultérieures, si vos recettes dépendent d'autres livres, vous devez inclure les instructions `depends` correspondantes dans le fichier `metadata.rb` de votre livre de recettes. Par exemple, si le livre inclut une recette avec une instruction telle que `include_recipe anothercookbook::somerecipe`, le fichier `metadata.rb` de votre livre de recettes doit comporter la ligne suivante : `depends "anothercookbook"`. Pour plus d'informations, consultez À propos des métadonnées des livres de recettes.

Les modèles doivent être dans un sous-répertoire du répertoire `templates` qui contient au moins un, voire plusieurs sous-répertoires. Ces sous-répertoires peuvent également contenir des sous-répertoires.

- Les modèles ont généralement un sous-répertoire `default` qui contient les fichiers de modèle utilisés par Chef par défaut.
- other (autre) représente les sous-répertoires facultatifs qui peuvent être utilisés pour des modèles de système d'exploitation spécifiques.
- Chef utilise automatiquement le modèle du sous-répertoire approprié, en fonction des conventions d'affectation de noms qui sont décrites dans File Specificity. Par exemple, pour les systèmes d'exploitation Linux et , vous pouvez mettre modèles spécifiques au système d'exploitation dans les sous-répertoires nommés `amazon`amazon ou `ubuntu`ubuntu, respectivement.

Les détails de gestion des livres de recettes personnalisés dépendent de votre type de référentiel préféré.

Pour utiliser une archive

1. Implémentez vos livres de recettes en utilisant la structure de dossier indiquée dans la section précédente.
2. Créez une archive compressée et chargez-la dans un compartiment Amazon S3 ou sur un site Web.

   Si vous mettez à jour vos livres de recettes, vous devez créer et télécharger un nouveau fichier d'archive. Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Pour utiliser un SCM

1. Configurez un référentiel Subversion ou Git à l'aide de la structure illustrée précédemment.
2. Le cas échéant, utilisez les fonctionnalités de contrôle de version du référentiel pour mettre en place plusieurs branches ou versions.

   Si vous mettez à jour vos livres de recettes, vous pouvez le faire dans une nouvelle branche et simplement demander OpsWorks à utiliser la nouvelle version. Vous pouvez aussi spécifier des versions balisées spécifiques. Pour en savoir plus, consultez Définition d'un référentiel de livres de recettes personnalisé.

Installation de livres de recettes personnalisésdécrit comment faire en sorte que AWS OpsWorks Stacks installe votre référentiel de livres de recettes sur les instances de la pile.

> **Important**
>
> Après avoir mis à jour les livres de recettes existants dans le référentiel, vous devez exécuter la commande `update_cookbooks` stack pour demander à AWS OpsWorks Stacks de mettre à jour le cache local de chaque instance en ligne. Pour de plus amples informations, veuillez consulter Exécution des commandes de pile.

Versions Chef

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks prend en charge plusieurs versions de Chef. Vous sélectionnez la version lorsque vous créez la pile. AWS OpsWorks Stacks installe ensuite cette version de Chef sur toutes les instances de la pile ainsi qu'un ensemble de recettes intégrées compatibles avec cette version. Si

vous installez des recettes personnalisées, elles doivent être compatibles avec la version Chef de la pile.

AWS OpsWorks Stacks prend actuellement en charge les versions 12, 11.10, 11.4 et 0.9 de Chef pour les stacks Linux, et Chef 12.2 (actuellement Chef 12.22) pour les stacks Windows. Pour plus de commodité, elles sont généralement désignées par leurs numéros de version majeure et de version mineure. Pour les piles Linux, vous pouvez utiliser le Gestionnaire de configuration et spécifier la version Chef à utiliser quand vous créez une pile. Les piles Windows doivent utiliser Chef 12.2. Pour plus d'informations, y compris les instructions sur la migration des piles vers des versions Chef plus récentes, consultez Versions Chef. Pour obtenir les informations complètes sur les versions, consultez AWS OpsWorks Systèmes d'exploitation Stacks.

Chef 12,2

Le support de Chef 12.2 a été introduit en mai 2015 et n'est utilisé que par Windows Stacks. La version actuelle de Chef sur les piles Windows est Chef 12.22. Il s'exécute avec Ruby 2.3.6 et utilise chef-client en mode local, lequel lance un serveur Chef local en mémoire appelé chef-zero. La présence de ce serveur permet aux recettes d'utiliser les conteneurs de données et la recherche Chef. La prise en charge présente certaines limites, décrites dans Mise en œuvre de recettes : Chef 12.2, mais vous pouvez exécuter la plupart des livres de recettes de la communauté sans modification.

Chef 12

Le support Chef 12 est disponible depuis décembre 2015 et n'est utilisé que par les piles Linux. Il s'exécute avec Ruby 2.1.6 ou 2.2.3 et utilise chef-client en mode local, ce qui permet aux recettes d'utiliser les conteneurs de données et la recherche Chef. Pour de plus amples informations, veuillez consulter AWS OpsWorks Systèmes d'exploitation Stacks.

Chef 11.10

Le support Chef 11.10 est disponible depuis mars 2014 et n'est utilisé que par les piles Linux. Il s'exécute avec Ruby 2.0.0 et utilise chef-client en mode local, ce qui permet aux recettes d'utiliser les conteneurs de données et la recherche Chef. La prise en charge présente certaines limites, décrites dans Mise en œuvre des recettes : Chef 11.10, mais vous pouvez exécuter la plupart des livres de recettes de la communauté sans modification. Vous pouvez également utiliser Berkshelf pour gérer les dépendances de votre livre de recettes. Les versions Berkshelf prises en charge dépendent du système d'exploitation. Pour de plus amples informations, veuillez consulter AWS OpsWorks Systèmes d'exploitation Stacks. Vous ne pouvez pas créer de piles CentOS qui utilisent Chef 11.10.

Chef 11.4

Le support Chef 11.4 est disponible depuis juillet 2013 et n'est utilisé que par les piles Linux. Il s'exécute avec Ruby 1.8.7 et utilise chef-solo, qui ne prend pas en charge les conteneurs de données ni la recherche Chef. Vous pouvez souvent utiliser des livres de recettes communautaires qui dépendent de ces fonctionnalités avec AWS OpsWorks Stacks, mais vous devez les modifier comme décrit dans. Migration vers une nouvelle version de Chef Vous ne pouvez pas créer de piles CentOS qui utilisent Chef 11.4. Les piles Chef 11.4 ne sont pas prises en charge sur les points de terminaison régionaux situés en dehors de la région USA Est (Virginie du Nord).

Chef 0.9

Chef 0.9 est utilisé uniquement par les piles Linux et n'est plus pris en charge. Notez ces informations :

- Vous ne pouvez pas utiliser la console pour créer une pile Chef 0.9.

  Vous devez utiliser l'interface de ligne ou l'API, ou vous devez créer une pile avec une version différente de Chef, puis modifier la configuration de la pile.
- Les nouvelles fonctionnalités de AWS OpsWorks Stacks ne sont pas disponibles pour les stacks Chef 0.9.
- Les nouvelles versions de système d'exploitation ne fournissent qu'une prise en charge limitée des piles Chef 0.9.

  En particulier, Amazon Linux 2014.09 et les versions ultérieures ne prennent pas en charge les piles Chef 0.9 avec des couches Rails App Server qui dépendent de Ruby 1.8.7.
- Les nouvelles régions AWS, y compris l'Europe (Francfort), ne prennent pas en charge les stacks Chef 0.9.

> **Note**
>
> Il est déconseillé d'utiliser Chef 0.9 pour les nouvelles piles. Vous devez migrer les piles existantes vers la dernière version Chef dès que possible.

Si vous souhaitez utiliser des livres de recettes communautaires avec AWS OpsWorks Stacks, nous vous recommandons de spécifier Chef 12 pour les nouvelles piles Linux et de migrer vos piles Linux existantes vers Chef 12. Vous pouvez utiliser la console, l'API ou la CLI AWS OpsWorks Stacks pour

migrer vos stacks existants vers une version plus récente de Chef. Pour de plus amples informations, veuillez consulter Migration vers une nouvelle version de Chef.

Rubriques

- Mise en œuvre de recettes pour Chef 12.2 Stacks
- Mise en œuvre des recettes pour les piles Chef 12
- Mise en œuvre des recettes pour les piles Chef 11.10
- Mise en œuvre des recettes pour les piles Chef 11.4
- Migration d'une pile Linux existante vers une nouvelle version de Chef

Mise en œuvre de recettes pour Chef 12.2 Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Chef 12.2 (actuellement Chef 12.22) est disponible uniquement sur les piles Windows, qui doivent exécuter cette version Chef.

- Les recettes doivent utiliser les ressources et les attributs propres à Windows à certaines fins.

  Pour plus d'informations, consultez Chef pour Microsoft Windows.
- Comme les exécutions de Chef utilisent Ruby 2.3.6, vos recettes peuvent utiliser la nouvelle syntaxe Ruby.
- Les recettes peuvent utiliser les conteneurs de données et la recherche Chef.

  Les piles Chef 12.2 peuvent utiliser de nombreux livres de cuisine communautaires sans modification. Pour plus d'informations, consultez Utilisation de Chef Search et Utilisation des conteneurs de données.

- La plupart des attributs de configuration et de déploiement de pile décrits dans AWS OpsWorks Référence du sac de données Stacks et Attributs des livres de recettes intégrés sont disponibles pour les recettes Windows.

  Vous pouvez utiliser la recherche Chef pour obtenir les valeurs des attributs. Pour obtenir un exemple, consultez Obtention des valeurs d'attribut avec la recherche de Chef. Pour obtenir la liste des attributs, consultez AWS OpsWorks Référence du sac de données Stacks.

Mise en œuvre des recettes pour les piles Chef 12

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les piles Chef 12 offrent les avantages suivants sur les piles Chef 11.10 :

- Comme les exécutions de Chef utilisent Ruby 2.1.6, vos recettes peuvent utiliser la nouvelle syntaxe Ruby.
- Les piles Chef 12 peuvent utiliser encore de plus de livres de recettes de la communauté sans modification. Sans livres de recettes intégrés sur le chemin, il n'y aura plus aucune risque de conflits de noms entre les livres de recettes intégrés et les livres de recettes personnalisés.
- Vous n'êtes plus limité aux versions de Berkshelf pour lesquelles AWS OpsWorks Stacks a fourni des packages prédéfinis. Berkshelf n'est plus installé sur les instances AWS OpsWorks Stacks dans Chef 12. Vous pouvez utiliser à la place n'importe quelle version Berkshelf sur votre ordinateur local.
- Il existe désormais une distinction claire entre les livres de recettes intégrés fournis par AWS OpsWorks Stacks avec Chef 12 (Elastic Load Balancing, Amazon RDS et Amazon ECS) et les livres de recettes personnalisés. Le dépannage des exécutions de Chef ayant échoué s'en trouve facilité.

Mise en œuvre des recettes pour les piles Chef 11.10

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les piles Chef 11.10 offrent les avantages suivants sur les piles Chef 11.4 :

- Comme les exécutions de Chef utilisent Ruby 2.0.0, vos recettes peuvent utiliser la nouvelle syntaxe Ruby.
- Les recettes peuvent utiliser les conteneurs de données et la recherche Chef.

  Les piles Chef 11.10 peuvent utiliser plusieurs livres de recettes de la communauté sans modification.
- Vous pouvez utiliser Berkshelf pour gérer les livres de recettes.

  Berkshelf fournit une manière beaucoup plus souple pour gérer vos livres de recettes personnalisés et utiliser les livres de recettes de la communauté dans une pile.
- Les livres de recettes doivent déclarer les dépendances dans `metadata.rb`.

  Si votre livre de recettes dépend d'un autre livre, vous devez inclure cette dépendance dans le fichier `metadata.rb` de votre livre de recettes. Par exemple, si le livre inclut une recette avec une instruction telle que `include_recipe anothercookbook::somerecipe`, le fichier `metadata.rb` de votre livre de recettes doit comporter la ligne suivante : `depends "anothercookbook"`.
- AWS OpsWorks Stacks installe un client MySQL sur les instances d'une pile uniquement si celle-ci inclut une couche MySQL.
- AWS OpsWorks Stacks installe un client Ganglia sur les instances d'une pile uniquement si celle-ci inclut une couche Ganglia.
- Si un déploiement exécute `bundle install` et que l'installation échoue, le déploiement échoue aussi.

> **Important**
>
> Ne réutilisez pas les noms des livres de recettes intégrés comme livres personnalisés ou de la communauté. Les livres de recettes personnalisés ayant le même nom que les livres intégrés risquent d'échouer. Pour une liste complète des livres de recettes intégrés disponibles avec les piles Chef 11.10, 11.4 et 0.9, consultez le référentiel opsworks-cookbooks sur. GitHub
>
> Les livres de recettes ayant des caractères non-ASCII et qui s'exécutent avec succès sur les piles Chef 0.9 et 11.4 peuvent échouer sur une pile Chef 11.10. La raison en est que les piles Chef 11.10 utilisent Ruby 2.0.0 pour les exécutions de Chef, qui est beaucoup plus strict sur l'encodage que Ruby 1.8.7. Pour garantir que ces livres s'exécutent correctement sur les piles Chef 11.10, chaque fichier utilisant des caractères non-ASCII doit comporter un commentaire fournissant une indication sur l'encodage. Par exemple, pour l'encodage UTF-8, le commentaire sera `# encoding: UTF-8`. Pour plus d'informations sur l'encodage Ruby 2.0.0, consultez Encodage.

Rubriques

- Installation et priorité des livres de recettes
- Utilisation de Chef Search
- Utilisation des conteneurs de données
- Utilisation de Berkshelf

Installation et priorité des livres de recettes

La procédure d'installation des livres de recettes AWS OpsWorks Stacks fonctionne quelque peu différemment pour les piles Chef 11.10 par rapport aux versions antérieures de Chef. Pour les piles Chef 11.10, une fois que AWS OpsWorks Stacks a installé les livres de recettes intégrés, personnalisés et Berkshelf, il les fusionne dans un répertoire commun dans l'ordre suivant :

1. Livres de recettes intégrés.
2. Livres de recettes Berkshelf, le cas échéant.
3. Livres de recettes personnalisés, le cas échéant.

Lorsque AWS OpsWorks Stacks effectue cette fusion, il copie l'intégralité du contenu des répertoires, y compris les recettes. En cas de doublons, les règles suivantes s'appliquent :

- Le contenu des livres de recettes Berkshelf a priorité sur les livres de recettes intégrés.
- Le contenu des livres de recettes personnalisés a priorité sur les livres de recettes Berkshelf.

Pour illustrer le fonctionnement de ce processus, envisagez le scénario suivant, où les trois répertoires de livres de recettes incluent tous un livre de recettes nommé `mycookbook` :

- Livres de recettes intégrés : `mycookbook` inclut un fichier d'attributs nommé`someattributes.rb`, un fichier modèle nommé `sometemplate.erb` et une recette nommée`somerecipe.rb`.
- Livres de cuisine Berkshelf - `mycookbook` comprend et. `sometemplate.erb` `somerecipe.rb`
- Livres de cuisine personnalisés — `mycookbook` inclus`somerecipe.rb`.

Le livre de recettes fusionné contient les éléments suivants :

- `someattributes.rb` du livre de recettes intégré.
- `sometemplate.erb` du livre de recettes Berkshelf.
- `somerecipe.rb` du livre de recettes personnalisé.

> **Important**
>
> Vous ne devez pas personnaliser votre pile Chef 11.10 en copiant la totalité d'un livre de recettes intégré sur votre espace de stockage, puis en modifiant des parties du livre. Une telle action remplace la totalité du livre de recettes intégré, recettes incluses. Si AWS OpsWorks Stacks met à jour ce livre de recettes, votre pile ne bénéficiera de ces mises à jour que si vous mettez à jour manuellement votre copie privée. Pour plus d'informations sur la personnalisation des piles, consultez Personnalisation des piles AWS OpsWorks.

Utilisation de Chef Search

Vous pouvez utiliser la méthode `search` de Chef dans vos recettes pour interroger les données de la pile. Vous utilisez la même syntaxe que pour le serveur Chef, mais AWS OpsWorks Stacks obtient les données à partir de l'objet du nœud local au lieu d'interroger un serveur Chef. Ces données comprennent :

- Les attributs de configuration et de déploiement de pile de l'instance.

- Les attributs des fichiers d'attributs des livres de recettes intégrés et personnalisés de l'instance.
- Données système collectées par Ohai.

Les attributs de configuration et de déploiement de la pile contiennent la plupart des informations que les recettes obtiennent généralement par le biais de la recherche, notamment des données telles que les noms d'hôtes et les adresses IP pour chaque instance en ligne de la pile. AWS OpsWorks Stacks met à jour ces attributs pour chaque événement du cycle de vie, ce qui garantit qu'ils reflètent précisément l'état actuel de la pile. Cela signifie que vous pouvez souvent utiliser dans votre pile des recettes de la communauté dépendant de la recherche sans modification. La méthode de recherche retourne toujours les données appropriées ; elles proviennent simplement des attributs de configuration et de déploiement de pile, et non d'un serveur.

La principale limite de AWS OpsWorks Stacks search est qu'elle ne gère que les données de l'objet du nœud local, en particulier la configuration de la pile et les attributs de déploiement. Pour cette raison, les types de données suivants ne peuvent pas être disponibles par le biais de la recherche :

- Attributs définis localement sur d'autres instances.

  Si une recette définit un attribut localement, ces informations ne sont pas renvoyées au service AWS OpsWorks Stacks. Vous ne pouvez donc pas accéder à ces données depuis d'autres instances en utilisant la recherche.
- Attributs `deploy` personnalisés.

  Vous pouvez spécifier le JSON personnalisé lorsque vous déployez une application et que les attributs correspondants sont installés sur les instances de la pile de ce déploiement. Cependant, si vous ne déployez que sur des instances sélectionnées, les attributs ne sont installés que sur ces instances. Les requêtes de ces attributs JSON personnalisés échouent sur toutes les autres instances. En outre, les attributs personnalisés ne sont inclus dans le JSON de configuration et de déploiement de la pile que pour ce déploiement. Ils ne sont accessibles que jusqu'à ce que le prochain événement du cycle de vie installe un nouvel ensemble d'attributs de déploiement et de configuration de la pile. Notez que si vous spécifiez un JSON personnalisé pour la pile, les attributs sont installés sur chaque instance pour tous les événements du cycle de vie et sont toujours accessibles par le biais de la recherche.
- Données Ohai des autres instances.

  L'outil Ohai de Chef obtient une grande variété de données système sur une instance et les ajoute à l'objet nœud. Comme ces données sont stockées localement et ne sont pas rapportées au

service AWS OpsWorks Stacks, la recherche ne peut pas accéder aux données Ohai à partir d'autres instances. Cependant, certaines de ces données peuvent être incluses dans les attributs de déploiement et de configuration de la pile.

- Instances hors connexion.

  Les attributs de configuration et de déploiement de la pile contiennent les données uniquement pour les instances en ligne.

L'extrait de recette suivant montre comment obtenir l'adresse IP privée de l'instance d'une couche PHP à l'aide de la recherche.

```
appserver = search(:node, "role:php-app").first
Chef::Log.info("The private IP is '#{appserver[:private_ip]}'")
```

> **Note**
>
> Lorsque AWS OpsWorks Stacks ajoute la configuration de la pile et les attributs de déploiement à l'objet du nœud, il crée en fait deux ensembles d'attributs de couche, chacun contenant les mêmes données. Un ensemble se trouve dans l'espace de `layers` noms, qui permet à AWS OpsWorks Stacks de stocker les données. L'autre ensemble se trouve dans l'espace de noms `role`, qui concerne la façon dont le serveur Chef stocke les données équivalentes. Le but de l'espace de `role` noms est de permettre au code de recherche implémenté pour le serveur Chef de s'exécuter sur une instance AWS OpsWorks Stacks. Si vous écrivez du code spécifiquement pour AWS OpsWorks Stacks, vous pouvez utiliser l'un `layers:php-app` ou `role:php-app` l'autre des exemples précédents et vous `search` obtiendrez le même résultat.

Utilisation des conteneurs de données

Vous pouvez utiliser la méthode `data_bag_item` de Chef dans vos recettes pour exécuter une requête sur un conteneur de données. Vous utilisez la même syntaxe comme vous le feriez pour un serveur Chef, mais AWS OpsWorks Stacks obtient les données des attributs de déploiement et de configuration de la pile de l'instance. Cependant, AWS OpsWorks Stacks ne prend pas actuellement en charge les environnements Chef et revient `_default` donc `node.chef_environment` toujours.

Vous créez un conteneur de données en utilisant un JSON personnalisé pour ajouter un ou plusieurs attributs à l'attribut `[:opsworks][:data_bags]`. L'exemple suivant illustre le format général de la création d'un conteneur de données dans un JSON personnalisé.

> **Note**
>
> Vous ne pouvez pas créer un conteneur de données en l'ajoutant à votre référentiel de livres de recettes. Vous devez utiliser un JSON personnalisé.

```
{
  "opsworks": {
    "data_bags": {
      "bag_name1": {
        "item_name1: {
          "key1" : “value1”,
          "key2" : “value2”,
          ...
        }
      },
      "bag_name2": {
        "item_name1": {
          "key1" : “value1”,
          "key2" : “value2”,
          ...
        }
      },
      ...
    }
  }
}
```

Généralement, vous spécifiez un JSON personnalisé pour la pile, lequel installe les attributs personnalisés sur chaque instance pour chaque événement ultérieur du cycle de vie. Vous pouvez aussi spécifier un JSON personnalisé lorsque vous déployez une application, mais ces attributs ne sont installés que pour ce déploiement et ne peuvent être installés que sur un ensemble sélectionné d'instances. Pour de plus amples informations, veuillez consulter Déploiement d'applications.

L'exemple de JSON personnalisé suivant crée un conteneur de données nommé `myapp`. Il possède un seul élément, `mysql`, avec deux paires clé-valeur.

```
{ "opsworks": {
    "data_bags": {
      "myapp": {
        "mysql": {
          "username": "default-user",
          "password": "default-pass"
        }
      }
    }
  }
}
```

Pour utiliser les données dans votre recette, vous pouvez appeler `data_bag_item` et lui transmettre les noms du conteneur de données et des valeurs, comme illustré dans l'extrait suivant.

```
mything = data_bag_item("myapp", "mysql")
Chef::Log.info("The username is '#{mything['username']}' ")
```

Pour modifier les données du conteneur de données, il suffit de modifier le JSON personnalisé et il sera installé sur les instances de la pile lors du prochain événement du cycle de vie.

Utilisation de Berkshelf

Avec les piles Chef 0.9 et Chef 11.4, vous ne pouvez installer qu'un seul référentiel de livres de recettes personnalisés. Avec les piles Chef 11.10, vous pouvez utiliser Berkshelf pour gérer vos livres de recettes et leurs dépendances, ce qui vous permet d'installer des livres de recettes à partir de plusieurs référentiels. (Pour plus d'informations, consultez Empaquetage local des dépendances des livres de recettes.) Avec Berkshelf, vous pouvez notamment installer des livres de cuisine communautaires AWS OpsWorks compatibles avec Stacks directement à partir de leurs référentiels au lieu de devoir les copier dans votre référentiel de livres de recettes personnalisé. Les versions Berkshelf prises en charge dépendent du système d'exploitation. Pour de plus amples informations, veuillez consulter AWS OpsWorks Systèmes d'exploitation Stacks.

Pour utiliser Berkshelf, vous devez l'activer explicitement, comme décrit dans Installation de livres de recettes personnalisés. Ensuite, incluez un fichier `Berksfile` dans le répertoire racine de votre référentiel de livres de recettes qui spécifie les livres à installer.

Pour spécifier une source de livres de recettes externe dans un fichier Berksfile, incluez un attribut source en haut du fichier qui spécifie l'URL du référentiel par défaut. Berkshelf recherchera les livres de recettes dans la source, URLs sauf si vous spécifiez explicitement un dépôt. Ensuite, incluez une ligne pour chaque livre de recettes que vous voulez installer dans le format suivant :

```
cookbook 'cookbook_name', ['>= cookbook_version'], [cookbook_options]
```

Les champs après `cookbook` spécifient le livre de recettes.

- *`cookbook_name`*— (Obligatoire) Spécifie le nom du livre de recettes.

  Si vous n'incluez aucun autre champ, Berkshelf installe le livre de recettes à partir de la source spécifiée. URLs
- *`cookbook_version`*— (Facultatif) Spécifie la ou les versions du livre de recettes.

  Vous pouvez utiliser un préfixe comme = ou >= pour spécifier une version ou une plage particulière de versions acceptables. Si vous ne spécifiez pas de version, Berkshelf installe la plus récente.
- *`cookbook_options`*— (Facultatif) Le dernier champ est un hachage contenant une ou plusieurs paires clé-valeur qui spécifient des options telles que l'emplacement du référentiel.

  Par exemple, vous pouvez inclure une clé `git` pour définir un référentiel Git particulier et une clé `tag` pour définir une branche de référentiel particulière. La spécification de la branche du référentiel est généralement le meilleur moyen de s'assurer que vous installez votre livre de recettes préféré.

> **Important**
>
> Ne déclarez pas de livres de recettes en incluant une ligne `metadata` dans votre fichier Berksfile et en déclarant les dépendances des livres de recettes dans `metadata.rb`. Pour un bon fonctionnement, les deux fichiers doivent se trouver dans le même répertoire. Avec AWS OpsWorks Stacks, le Berksfile doit se trouver dans le répertoire racine du dépôt, mais les `metadata.rb` fichiers doivent se trouver dans leurs répertoires de livres de recettes respectifs. Vous devez à la place déclarer explicitement les livres de recettes externes dans le fichier Berksfile.

Voici un exemple de fichier Berksfile qui indique les différentes façons de spécifier les livres de recettes. Pour plus d'informations sur la façon de créer un fichier Berksfile, consultez 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'
```

Le fichier installe les livres de recettes suivants :

- La version `apt` la plus récente à partir du référentiel des livres de recettes de la communauté.
- La version `bluepill` la plus récente à partir des livres de recettes de la communauté, aussi longtemps qu'il s'agit de la version 2.3.1 ou ultérieure.
- La version `ark` la plus récente à partir d'un référentiel spécifié.

  L'URL de cet exemple est celle d'un dépôt de livres de recettes communautaire public sur GitHub, mais vous pouvez installer des livres de recettes provenant d'autres référentiels, y compris des référentiels privés. Pour plus d'informations, consultez Berkshelf.
- Le livre de recettes `build-essential` à partir de la branche v1.4.2 du référentiel spécifié.

Un référentiel de livres de recettes personnalisés peut contenir des livres de recettes personnalisés en plus d'un fichier Berksfile. Dans ce cas, AWS OpsWorks Stacks installe les deux ensembles de livres de recettes, ce qui signifie qu'une instance peut avoir jusqu'à trois référentiels de livres de recettes.

- Les livres de recettes intégrés sont installés dans `/opt/aws/opsworks/current/cookbooks`.
- Si votre référentiel de livres de recettes personnalisés contient des livres de recettes, ils sont installés dans `/opt/aws/opsworks/current/site-cookbooks`.
- Si vous avez activé Berkshelf et que votre référentiel de livres de recettes personnalisés contient un fichier Berksfile, les livres de recettes spécifiés sont installés dans `/opt/aws/opsworks/current/berkshelf-cookbooks`.

Les livres de recettes intégrés et vos livres de recettes personnalisés sont installés sur chaque instance lors de l'installation et ne sont pas mis à jour ultérieurement, sauf si vous exécutez manuellement la commande Update Custom Cookbooks stack. AWS OpsWorks Stacks fonctionne `berks install` pour chaque course de Chef. Vos livres de recettes Berkshelf sont donc mis à jour pour chaque événement du cycle de vie, conformément aux règles suivantes :

- Si vous avez une nouvelle version du livre de recettes dans le référentiel, cette opération met à jour le livre à partir du référentiel.
- Sinon, cette opération met à jour les livres de recettes Berkshelf à partir d'un cache local.

> **Note**
>
> Comme l'opération remplace les livres de recettes Berkshelf, si vous avez modifié les copies locales de livres de recettes, les modifications sont écrasées. Pour plus d'informations, consultez Berkshelf.

Vous pouvez aussi mettre à jour vos livres de recettes Berkshelf en exécutant la commande de pile Update Custom Cookbooks, qui met à jour aussi bien les livres Berkshelf que vos livres de recettes personnalisés.

Mise en œuvre des recettes pour les piles Chef 11.4

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Important**
>
> Ne réutilisez pas les noms des livres de recettes intégrés comme livres personnalisés ou de la communauté. Les livres de recettes personnalisés ayant le même nom que les livres intégrés risquent d'échouer. Pour une liste complète des livres de recettes intégrés

disponibles avec les piles Chef 11.10, 11.4 et 0.9, consultez le référentiel opsworks-cookbooks sur. GitHub

La principale limitation des piles Chef 11.4 est que les recettes ne peuvent utiliser la recherche ou les conteneurs de données Chef. Cependant, AWS OpsWorks Stacks installe sur chaque instance des attributs de configuration et de déploiement de la pile qui contiennent la plupart des informations que vous pourriez obtenir grâce à la recherche, notamment les suivantes :

- Données définies par l'utilisateur à partir de la console, telles que les noms d'hôte ou d'application.
- Les données de configuration de la pile générées par le service AWS OpsWorks Stacks, telles que les couches, les applications et les instances de la pile, ainsi que les détails relatifs à chaque instance, tels que l'adresse IP.
- Attributs JSON personnalisés qui contiennent les données fournies par l'utilisateur et peuvent servir pratiquement le même objectif que les conteneurs de données.

AWS OpsWorks Stacks installe une version actuelle de la configuration de la pile et des attributs de déploiement sur chaque instance pour chaque événement du cycle de vie, avant de démarrer l'exécution Chef de l'événement. Les données sont disponibles pour les recettes via la syntaxe `node[:attribute][:child_attribute][...]` standard. Par exemple, les attributs de configuration et de déploiement de la pile incluent le nom de la pile, `node[:opsworks][:stack][:name]`.

L'extrait suivant de l'une des recettes intégrées obtient le nom de la pile et l'utilise pour créer un fichier de configuration.

```
template '/etc/ganglia/gmetad.conf' do
  source 'gmetad.conf.erb'
  mode '0644'
  variables :stack_name => node[:opsworks][:stack][:name]
  notifies :restart, "service[gmetad]"
end
```

La plupart des valeurs des attributs de configuration et de déploiement de la pile contiennent plusieurs attributs. Vous devez itérer sur ces attributs afin d'obtenir les informations dont vous avez besoin. L'exemple suivant montre un extrait des attributs de configuration et de déploiement de la pile, représentés sous la forme d'un objet JSON pour plus de commodité. Il possède un attribut de

niveau supérieur, `deploy`, qui contient un attribut pour chacune des applications de la pile, nommé d'après le nom court de l'application.

```
{
  ...
  "deploy": {
    "app1_shortname": {
      "document_root": "app1_root",
      "deploy_to": "deploy_directory",
      "application_type": "php",
      ...
    },
    "app2_shortname": {
      "document_root": "app2_root",
      ...
    }
  },
  ...
}
```

Chaque attribut d'application contient un ensemble d'attributs qui caractérisent l'application. Par exemple, l'attribut `deploy_to` représente le répertoire de déploiement de l'application. L'extrait suivant définit l'utilisateur, le groupe et le chemin d'accès du répertoire de déploiement de chaque application.

```
node[:deploy].each do |application, deploy|
  opsworks_deploy_dir do
    user deploy[:user]
    group deploy[:group]
    path deploy[:deploy_to]
  end
  ...
end
```

Pour plus d'informations sur les attributs de configuration et de déploiement de la pile, consultez Personnalisation des piles AWS OpsWorks. Pour plus d'informations sur les répertoires de déploiement, consultez Recettes Deploy.

Les piles Chef 11.4 ne prennent pas en charge les conteneurs de données, mais vous pouvez arbitrairement ajouter des données aux attributs de configuration et de déploiement de la pile en spécifiant un JSON personnalisé. Vos recettes peuvent alors accéder aux données en utilisant la syntaxe de nœud Chef standard. Pour de plus amples informations, veuillez consulter Utilisation du JSON personnalisé.

Si vous avez besoin de la fonctionnalité d'un sac de données chiffré, l'une des options consiste à stocker les attributs sensibles dans un emplacement sécurisé tel qu'un compartiment Amazon S3 privé. Vos recettes peuvent ensuite utiliser le SDK AWS Ruby, qui est installé sur toutes les instances AWS OpsWorks Stacks, pour télécharger les données depuis le compartiment.

> **Note**
>
> Chaque instance AWS OpsWorks Stacks possède un profil d'instance. Le rôle IAM associé indique les ressources AWS auxquelles les applications qui s'exécutent sur l'instance peuvent accéder. Pour que vos recettes puissent accéder à un compartiment Amazon S3, la politique du rôle doit inclure une déclaration similaire à la suivante, qui accorde l'autorisation de récupérer des fichiers depuis un compartiment spécifié.
>
> ```
> "Action": ["s3:GetObject"],
> "Effect": "Allow",
> "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
> ```
>
> Pour plus d'informations sur les profils d'instance, consultez Spécification des autorisations pour les applications exécutées sur EC2 des instances.

Migration d'une pile Linux existante vers une nouvelle version de Chef

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez utiliser la console, l'API ou la CLI AWS OpsWorks Stacks pour migrer vos stacks Linux vers une version plus récente de Chef. Cependant, vos recettes peuvent nécessiter une modification pour être compatibles avec la version la plus récente. Lors de la préparation de la migration d'une pile, prenez en compte les points suivants.

- Vous ne pouvez pas modifier les versions de pile de AWS OpsWorks Stacks de Chef 11 à Chef 12 en modifiant ou en clonant la pile. Une mise à niveau de version majeure de Chef ne peut pas être effectuée à l'aide de la procédure décrite dans cette section. Pour plus d'informations sur le passage de Chef 11.10 à Chef 12, consultez Mise en œuvre des recettes : Chef 12.
- La transition d'une version de Chef à une autre implique un certain nombre de modifications, dont certaines nouveautés.

  Pour plus d'informations sur le passage de Chef 0.9 à Chef 11.4, consultez Migration vers une nouvelle version de Chef. Pour plus d'informations sur le passage de Chef 11.4 à Chef 11.10, consultez Mise en œuvre des recettes : Chef 11.10. Pour plus d'informations sur le passage de Chef 11.10 à Chef 12, consultez Mise en œuvre des recettes : Chef 12.
- Les exécutions de Chef utilisent une autre version de Ruby sur les piles Chef 0.9 et Chef 11.4 (Ruby 1.8.7), les piles Chef 11.10 (Ruby 2.0.0) et les piles Chef 12 (Ruby 2.1.6).

  Pour de plus amples informations, veuillez consulter Versions de Ruby.
- Les piles Chef 11.10 gèrent l'installation des livres de recettes différemment des piles Chef 0.9 ou Chef 11.4.

  Cette différence peut entraîner des problèmes lors de la migration de piles qui utilisent les livres de recettes personnalisés vers Chef 11.10. Pour de plus amples informations, veuillez consulter Installation et priorité des livres de recettes.

Les paragraphes suivants concernent les instructions recommandées pour la migration d'une pile Chef vers une version Chef plus récente :

Pour migrer une pile vers une version Chef plus récente

1. Clonez votre pile de production. Sur la page Clone Stack, cliquez sur Advanced>> pour afficher la section Configuration Management, puis remplacez Chef version par la version suivante la plus élevée.

> **Note**
>
> Si vous démarrez avec une pile Chef 0.9, vous ne pouvez pas procéder directement à sa mise à niveau vers Chef 11.10 ; vous devez d'abord effectuer une mise à niveau vers Chef 11.4. Si vous souhaitez migrer votre pile vers Chef 11.10 avant de tester vos recettes, attendez 20 minutes que la mise à jour soit terminée, puis mettez à niveau la pile de la version 11.4 à la version 11.10.

2. Ajoutez des instances aux couches et testez les applications et les livres de recettes de la pile clonée sur un système intermédiaire ou de test. Pour en savoir plus, consultez cette page consacrée à Chef.
3. Lorsque les résultats des tests sont satisfaisants, effectuez les opérations suivantes :

   - S'il s'agit de votre version de Chef souhaitée, vous pouvez utiliser la pile clonée comme pile de production ou réinitialiser la version Chef sur votre pile de production.
   - Si vous migrez une pile Chef 0.9 vers 11.10 en deux étapes, répétez le processus de migration de la pile de Chef 11.4 vers Chef 11.10.

> **Note**
>
> Lorsque vous testez des recettes, vous pouvez utiliser SSH pour vous connecter à l'instance, puis utiliser la commande run_command de la CLI de l'Instance Agent pour exécuter les recettes associées aux différents événements du cycle de vie. L'interface de ligne de commande de l'agent est particulièrement utile pour tester les recettes Setup, car vous pouvez l'utiliser même en cas de défaillance de Setup et si l'instance n'atteint pas l'état « en ligne ». Vous pouvez également utiliser la commande de pile Setup pour exécuter à nouveau les recettes Setup, mais cette commande est disponible uniquement si Setup a réussi et que l'instance est en ligne.

Il est possible de mettre à jour une pile en cours d'exécution avec une nouvelle version de Chef.

Pour mettre à jour une pile en cours d'exécution avec une nouvelle version de Chef

1. Modifiez la pile pour modifier le paramètre de pile Chef version.

2. Enregistrez les nouveaux paramètres et attendez que AWS OpsWorks Stacks mette à jour les instances, ce qui prend généralement 15 à 20 minutes.

> **Important**
>
> AWS OpsWorks Stacks ne synchronise pas la mise à jour de la version de Chef avec les événements du cycle de vie. Si vous souhaitez mettre à jour la version de Chef sur une pile de production, vous devez veiller à vous assurer que la mise à jour est terminée avant que le prochain événement du cycle de vie n'ait lieu. Si un événement se produit, généralement un événement Deploy ou Configure, l'agent d'instance met à jour vos livres de recettes personnalisés et exécute les recettes attribuées à l'événement, que la mise à jour de version soit terminée ou non. Il n'existe aucun moyen direct pour déterminer que la mise à jour de la version est terminée, mais les journaux du déploiement incluent la version de Chef.

Versions de Ruby

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Ruby est installé sur toutes les instances d'une pile Linux. AWS OpsWorks Stacks installe un package Ruby sur chaque instance, qu'il utilise pour exécuter les recettes Chef et l'agent d'instance. AWS OpsWorks Stacks détermine la version Ruby en fonction de la version de Chef exécutée par la pile. Ne tentez pas de modifier cette version, sans quoi vous pourriez désactiver l'agent de l'instance.

AWS OpsWorks Stacks n'installe pas d'application exécutable Ruby sur Windows Stacks. Le client Chef 12.2 est fourni avec Ruby 2.0.0 p451, mais l'exécutable Ruby n'est pas ajouté à la variable d'environnement PATH des instances. Si vous souhaitez utiliser ce fichier exécutable pour exécuter le code Ruby, il se trouve dans `\opscode\chef\embedded\bin\ruby.exe` sur votre disque Windows.

Le tableau suivant récapitule les versions de AWS OpsWorks Stacks Ruby. Les versions Ruby disponibles de l'application dépendent aussi du système d'exploitation de l'instance. Pour plus d'informations, y compris les versions de correctifs disponibles, consultez AWS OpsWorks Systèmes d'exploitation Stacks.

Version de Chef	Version Ruby Chef	Versions Ruby de l'application disponibles
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	Aucune
12.22 (d)	2.3.6	Aucun

(a) N'est pas disponible avec Amazon Linux 2014.09 et version ultérieure, Red Hat Enterprise Linux (RHEL) ou Ubuntu 14.04 LTS.

(b) Disponible uniquement sur les piles Linux.

(c) N'est pas disponible avec RHEL.

(d) Seulement disponible sur les piles Windows. La version majeure est 12.2. La version mineure actuelle est 12.22.

(e) L'obsolescence est effective, le support a pris fin.

Les emplacements d'installation dépendent de la version Chef :

- Les applications utilisent l'exécutable `/usr/local/bin/ruby` pour toutes les versions Chef.
- Pour Chef 0.9 et 11.4, l'agent de l'instance et les recettes Chef utilisent l'exécutable `/usr/bin/ruby`.
- Pour Chef 11.10, l'agent de l'instance et les recettes Chef utilisent l'exécutable `/opt/aws/opsworks/local/bin/ruby`.

Installation de livres de recettes personnalisés

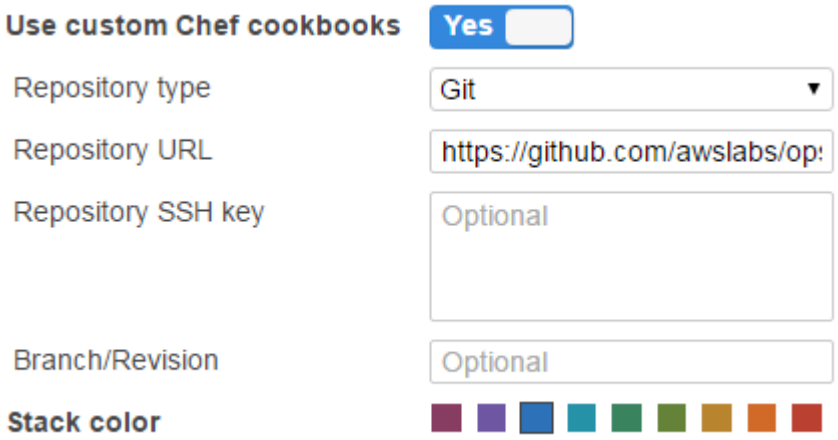
> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Pour qu'une pile installe et utilise des livres de recettes personnalisés, vous devez configurer la pile de façon à ce qu'elle accepte les livres personnalisés, si ce n'est déjà fait. Vous devez ensuite fournir l'URL de référentiel et les informations connexes, par exemple un mot de passe.

> **Important**
>
> Une fois que vous avez configuré la pile pour prendre en charge les livres de recettes personnalisés, AWS OpsWorks Stacks installe automatiquement vos livres de recettes sur toutes les nouvelles instances au démarrage. Cependant, vous devez explicitement demander à AWS OpsWorks Stacks d'installer des livres de recettes nouveaux ou mis à jour sur toutes les instances existantes en exécutant la commande Update Custom Cookbooks stack. Pour de plus amples informations, veuillez consulter Mise à jour des livres de recettes personnalisés. Avant d'activer Use custom Chef cookbooks (Utiliser les livres de recettes Chef personnalisés) sur votre pile, assurez-vous que les livres de recettes personnalisés et de la communauté que vous exécutez prennent en charge la version de Chef utilisée par votre pile.

Pour configurer une pile de livres de recettes personnalisés

1. Sur la page de votre pile, cliquez sur Stack Settings (Paramètres de la pile) pour afficher la page Settings (Paramètres). Cliquez sur Edit (Modifier) pour modifier les paramètres.
2. Basculer Use custom Chef cookbooks (Utiliser les livres de recettes Chef personnalisés) sur Yes (Oui).



3. Configurez vos livres de recettes personnalisés.

Lorsque vous avez terminé, cliquez sur Save (Enregistrer) pour enregistrer la pile mise à jour.

Définition d'un référentiel de livres de recettes personnalisé

Les piles Linux peuvent installer des livres de recettes personnalisés à partir de l'un des types de référentiels suivants :

- Archives HTTP ou Amazon S3.

  Elles peuvent être publiques ou privées, mais Amazon S3 est généralement l'option préférée pour les archives privées.
- Les référentiels Git et Subversion permettent de contrôler le code source et d'avoir plusieurs versions.

Windows Stacks peut installer des livres de recettes personnalisés à partir des archives Amazon S3 et des référentiels Git.

Tous les types de référentiels auront les champs obligatoires suivants.

- Type de référentiel : type de référentiel
- URL du référentiel : URL du référentiel

AWS OpsWorks Stacks prend en charge les sites de dépôt Git hébergés publiquement tels que GitHubBitbucket, ainsi que les serveurs Git hébergés en privé. Pour les référentiels Git, vous devez

utiliser l'un des formats d'URL suivants, selon que le référentiel est public ou privé. Suivez les mêmes instructions d'URL pour les sous-modules Git.

Pour un référentiel Git public, utilisez les protocoles HTTPS ou Git en lecture seule :

- Git en lecture seule —. `git://github.com/amazonwebservices/opsworks-example-cookbooks.git`
- HTTPS —`https://github.com/amazonwebservices/opsworks-example-cookbooks.git`.

Pour un référentiel Git privé, vous devez utiliser le format de lecture/écriture SSH, comme illustré dans les exemples suivants :

- Référentiels Github —. `git@github.com:`*`project`*`/`*`repository`*
- Référentiels sur un serveur Git — *`user`*`@`*`server`*`:`*`project`*`/`*`repository`*

Les autres paramètres varient selon le type de référentiel et sont décrits dans les sections suivantes.

Archive HTTP

Si vous sélectionnez Http Archive (Archive HTTP) pour Repository type (Type de référentiel), deux paramètres supplémentaires s'affichent et vous devez les remplir si l'archive est protégée par mot de passe.

- Nom d'utilisateur : votre nom d'utilisateur
- Mot de passe —Votre mot de passe

Archive Amazon S3

Le fait de sélectionner S3 Archive (Archive S3) comme Repository type (Type de référentiel) permet d'afficher les paramètres optionnels et supplémentaires suivants. AWS OpsWorks Stacks peut accéder à votre référentiel en utilisant les EC2 rôles Amazon (authentification du responsable du système d'exploitation hôte), que vous utilisiez l'API ou la AWS OpsWorks console Stacks.

- ID de clé d'accès : identifiant de clé d'accès AWS, tel queAKIAIOSFODNN7EXAMPLE.
- Clé d'accès secrète : clé d'accès secrète AWS correspondante, telle quewJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY.

Référentiel Git

Si vous sélectionnez Git sous Source Control (Contrôle de source), les paramètres facultatifs supplémentaires suivants s'affichent :

Repository SSH key (Clé SSH du référentiel)

Vous devez spécifier une clé SSH de déploiement pour accéder aux référentiels Git privés. Pour les sous-modules Git, la clé spécifiée doit avoir accès à ces sous-modules. Pour de plus amples informations, veuillez consulter Utilisation de clés SSH d'un référentiel Git.

> **Important**
>
> La clé SSH de déploiement ne peut pas nécessiter de mot de passe ; AWS OpsWorks Stacks n'a aucun moyen de le transmettre.

Branch/Revision

Si le dépôt comporte plusieurs branches, AWS OpsWorks Stacks télécharge la branche principale par défaut. Pour spécifier une branche en particulier, entrez le nom de la branche, le SHA1 hachage ou le nom de balise. Pour spécifier une validation particulière, saisissez l'ID de validation complet de 40 chiffres hexadécimaux.

Référentiel Subversion

Si vous sélectionnez Subversion sous Source Control (Contrôle de source), les paramètres supplémentaires suivants s'affichent :

- Nom d'utilisateur : votre nom d'utilisateur, pour les référentiels privés.
- Mot de passe : votre mot de passe, pour les dépôts privés.
- Révision — [Facultatif] Le nom de la révision, si vous avez plusieurs révisions.

Pour spécifier une branche ou une balise, vous devez modifier l'URL du référentiel, par exemple : **`http://repository_domain/repos/myapp/branches/my-apps-branch`** ou **`http://repository_domain_name/repos/calc/myapp/my-apps-tag`**.

Mise à jour des livres de recettes personnalisés

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Lorsque vous fournissez à AWS OpsWorks Stacks des livres de recettes personnalisés, les recettes de configuration intégrées créent un cache local sur chaque instance nouvellement démarrée et téléchargent les livres de recettes dans le cache. AWS OpsWorks Stacks exécute ensuite les recettes à partir du cache, et non du référentiel. Si vous modifiez les livres de recettes personnalisés dans le référentiel, vous devez vous assurer que les livres de recettes mis à jour sont installés dans les caches locaux de vos instances. AWS OpsWorks Stacks déploie automatiquement les derniers livres de recettes sur les nouvelles instances lorsqu'elles sont démarrées. Pour les instances existantes, cependant, la situation est différente :

- Vous devez déployer manuellement les livres de recettes personnalisés mis à jour sur les instances en ligne.
- Vous n'avez pas besoin de déployer les livres de recettes personnalisés mis à jour sur les instances basées sur le stockage d'instances hors connexion, y compris les instances basées sur les charges et sur le temps.

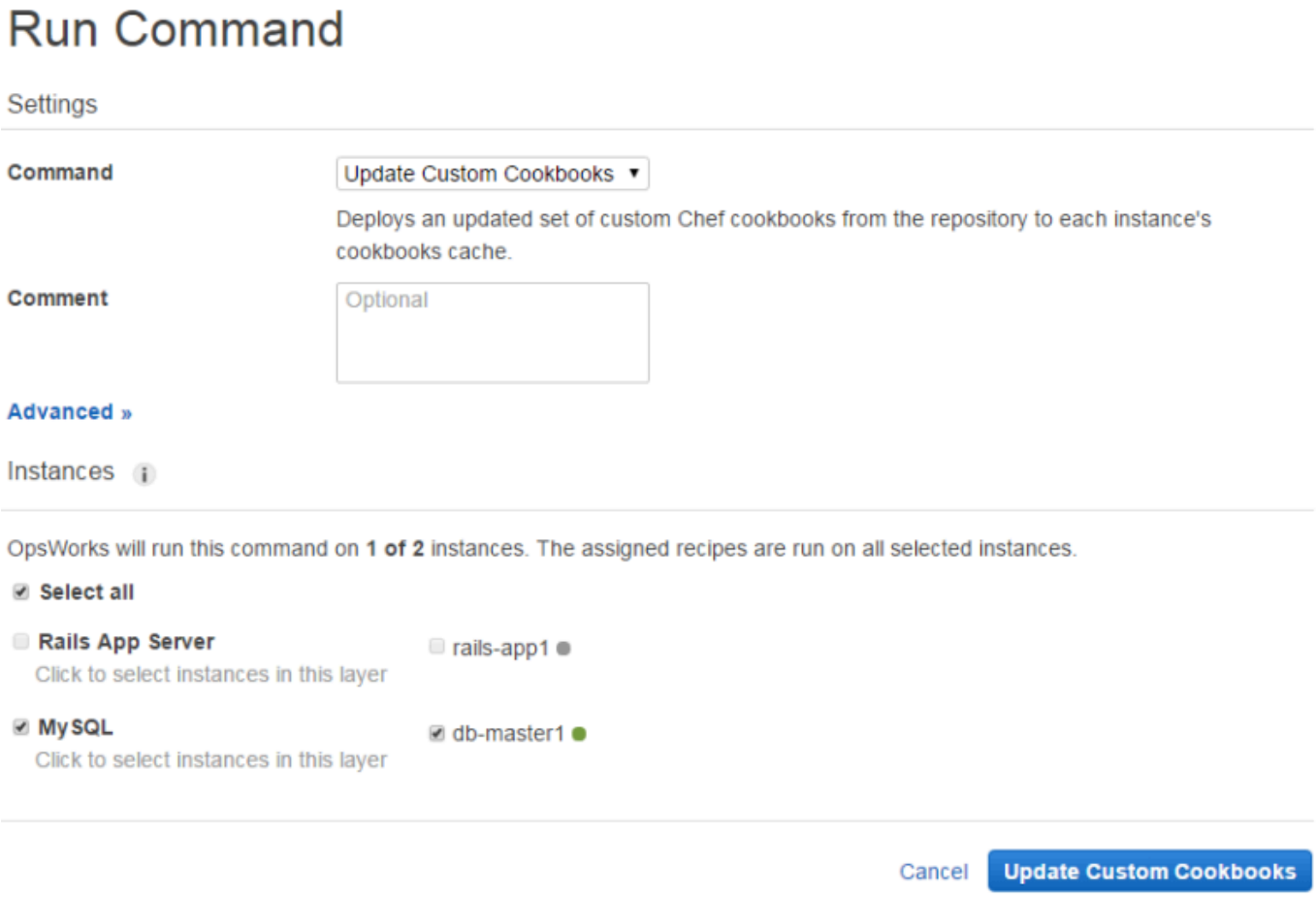
  AWS OpsWorks Stacks déploie automatiquement les livres de recettes actuels lorsque les instances sont redémarrées.
- Vous devez démarrer les instances 24/7 basées sur EBS qui ne reposent ni sur les charges ni sur le temps.
- Vous ne pouvez pas démarrer des instances hors connexion basées sur EBS et reposant sur les charges et le temps. L'approche la plus simple consiste donc à supprimer les instances hors connexion et à ajouter des instances pour les remplacer.

  Comme il s'agit désormais de nouvelles instances, AWS OpsWorks Stacks déploie automatiquement les livres de recettes personnalisés actuels au démarrage des instances.

Pour mettre à jour manuellement les livres de recettes personnalisés

1. Mettez à jour votre dépôt avec les livres de recettes modifiés. AWS OpsWorks Stacks utilise l'URL du cache que vous avez fournie lors de l'installation initiale des livres de recettes. Le nom du fichier racine du livre de recettes, l'emplacement du référentiel et les droits d'accès ne doivent donc pas changer.

   - Pour les référentiels Amazon S3 ou HTTP, remplacez le fichier .zip d'origine par un nouveau fichier .zip portant le même nom.
   - Pour les référentiels Git ou Subversion, modifiez vos paramètres de pile pour remplacer la valeur du champ Branch/Revision (Branche/Révision) par la nouvelle version.
2. Sur la page de la pile, cliquez sur Run Command (Exécuter la commande) et sélectionnez la commande Update Custom Cookbooks (Mettre à jour les livres de recettes personnalisées).



3. Ajoutez un commentaire si vous le souhaitez.
4. Spécifiez éventuellement un objet JSON personnalisé pour la commande afin d'ajouter des attributs personnalisés à la configuration de la pile et aux attributs de déploiement que AWS

OpsWorks Stacks installe sur les instances. Pour plus d'informations, consultez Utilisation du JSON personnalisé et Remplacement des attributs.

5. Par défaut, AWS OpsWorks Stacks met à jour les livres de recettes sur chaque instance. Pour spécifier les instances à mettre à jour, sélectionnez les instances appropriées dans la liste à la fin de la page. Pour sélectionner toutes les instances d'une couche, cochez la case de la couche appropriée dans la colonne de gauche.
6. Cliquez sur Mettre à jour les livres de recettes personnalisés pour installer les livres de recettes mis à jour. AWS OpsWorks Stacks supprime les livres de recettes personnalisés mis en cache sur les instances spécifiées et installe les nouveaux livres de recettes depuis le référentiel.

> **Note**
>
> Cette procédure est obligatoire uniquement pour les instances existantes qui ont d'anciennes versions des livres de recettes dans leurs caches. Si vous ajoutez ensuite des instances à une couche, AWS OpsWorks Stacks déploie les livres de recettes actuellement présents dans le référentiel afin qu'ils obtiennent automatiquement la dernière version.

Exécution des recettes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez exécuter des recettes de deux façons :

- Automatiquement, en attribuant des recettes à l'événement du cycle de vie de la couche appropriée.
- Manuellement, en choisissant la commande de pile Execute Recipes ou à l'aide de l'interface de ligne de commande de l'agent.

Rubriques

- AWS OpsWorks Événements de cycle de vie
- Exécution automatique des recettes
- Exécution manuelle des recettes

AWS OpsWorks Événements de cycle de vie

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Chaque couche a un ensemble de cinq événements de cycle de vie, chacune étant associée à un ensemble de recettes qui sont propres à la couche. Lorsqu'un événement se produit sur une instance de couche, AWS OpsWorks Stacks exécute automatiquement l'ensemble de recettes approprié. Pour fournir une réponse personnalisée à ces événements, implémentez des recettes personnalisées et attribuez-les aux événements appropriés pour chaque couche. AWS OpsWorks Stacks exécute ces recettes après les recettes intégrées à l'événement.

Setup

Cet événement se produit après le démarrage d'une instance. Vous pouvez également déclencher manuellement l'Setupévénement à l'aide de la commande Setup stack. AWS OpsWorks Stacks exécute des recettes qui configurent l'instance en fonction de sa couche. Par exemple, si l'instance est membre de la couche Rails App Server, les Setup recettes installent Apache, Ruby Enterprise Edition, Passenger et Ruby on Rails.

> **Note**
>
> Un événement Setup nécessite une instance hors service. Comme l'instance n'est pas à l'état Online quand les événements de cycle de vie Setup s'exécutent, les instances sur lesquelles vous exécutez Setup sont supprimées d'un équilibreur de charge.

Configure

Cet événement se produit sur toutes les instances de la pile dans les situations suivantes :

- Une instance entre ou quitte l'état en ligne.
- Vous associez une adresse IP Elastic à une instance ou vous en dissociez une d'une instance.
- Vous attachez un équilibreur de charge Elastic Load Balancing à une couche, ou vous en détachez un.

Par exemple, supposons que votre pile comporte des instances A, B et C et que vous en démarriez une nouvelle, D. Une fois que D a terminé d'exécuter ses recettes de configuration, AWS OpsWorks Stacks déclenche l'Configureévénement sur A, B, C et D. Si vous arrêtez A par la suite, AWS OpsWorks Stacks déclenche l'Configureévénement sur B, C et D. AWS OpsWorks Stacks répond à l'Configureévénement en exécutant les Configure recettes de chaque couche, qui mettent à jour la configuration des instances pour refléter l'ensemble actuel d'instances en ligne. L'événement Configure est donc le moment idéal pour régénérer les fichiers de configuration. Par exemple, les HAProxy Configure recettes reconfigurent l'équilibreur de charge pour tenir compte de toute modification apportée à l'ensemble des instances de serveurs d'applications en ligne.

Vous pouvez également déclencher manuellement l'événement Configure en utilisant la commande de pile Configure.

Deploy

Cet événement se produit lorsque vous exécutez une commande Deploy, généralement pour déployer une application dans un ensemble d'instances de serveurs d'application. Les instances exécutent des recettes qui déploient l'application et tous les fichiers associés depuis leur référentiel jusqu'aux instances de la couche. Par exemple, pour une instance de serveur d'applications Rails, les recettes Deploy extraient une application Ruby spécifiée et demandent à Phusion Passenger de la recharger. Vous pouvez également exécuter Deploy sur d'autres instances afin qu'elles puissent, par exemple, mettre à jour leur configuration et s'adapter à l'application nouvellement déployée.

> **Note**
>
> Setup inclut Deploy ; elle exécute les recettes Deploy une fois l'installation terminée.

Undeploy

Cet évènement se produit lorsque vous supprimez une application ou lorsque vous exécutez une commande Undeploy pour supprimer une application d'un ensemble d'instances de serveurs d'application. Les instances spécifiées exécutent des recettes pour supprimer toutes les versions de l'application et effectuer n'importe quel nettoyage requis.

Shutdown

Cet événement se produit une fois que vous avez demandé à AWS OpsWorks Stacks d'arrêter une instance, mais avant que l' EC2 instance Amazon associée ne soit réellement résiliée. AWS OpsWorks Stacks exécute des recettes pour effectuer des tâches de nettoyage telles que la fermeture de services.

Si vous avez attaché un équilibreur de charge Elastic Load Balancing à la couche et activé la prise en charge du drainage des connexions, AWS OpsWorks Stacks attend que le drainage des connexions soit terminé avant de déclencher l'événement. Shutdown

Après avoir déclenché un Shutdown événement, AWS OpsWorks Stacks accorde aux Shutdown recettes un délai spécifié pour effectuer leurs tâches, puis arrête ou met fin à l'instance Amazon EC2 . La valeur par défaut du délai d'attente de Shutdown est de 120 secondes. Si vos recettes Shutdown ont besoin de plus de temps, vous pouvez modifier la configuration de la couche pour modifier la valeur du délai d'attente. Pour plus d'informations sur l'opération Shutdown pour une instance, consultez Arrêt d'une instance.

> **Note**
>
> Le redémarrage d'une instance ne déclenche aucun événement du cycle de vie.

Pour plus d'informations sur les commandes d'application Deploy et Undeploy, consultez Déploiement d'applications.

Une fois qu'une instance a terminé son démarrage, la séquence de démarrage restante est la suivante :

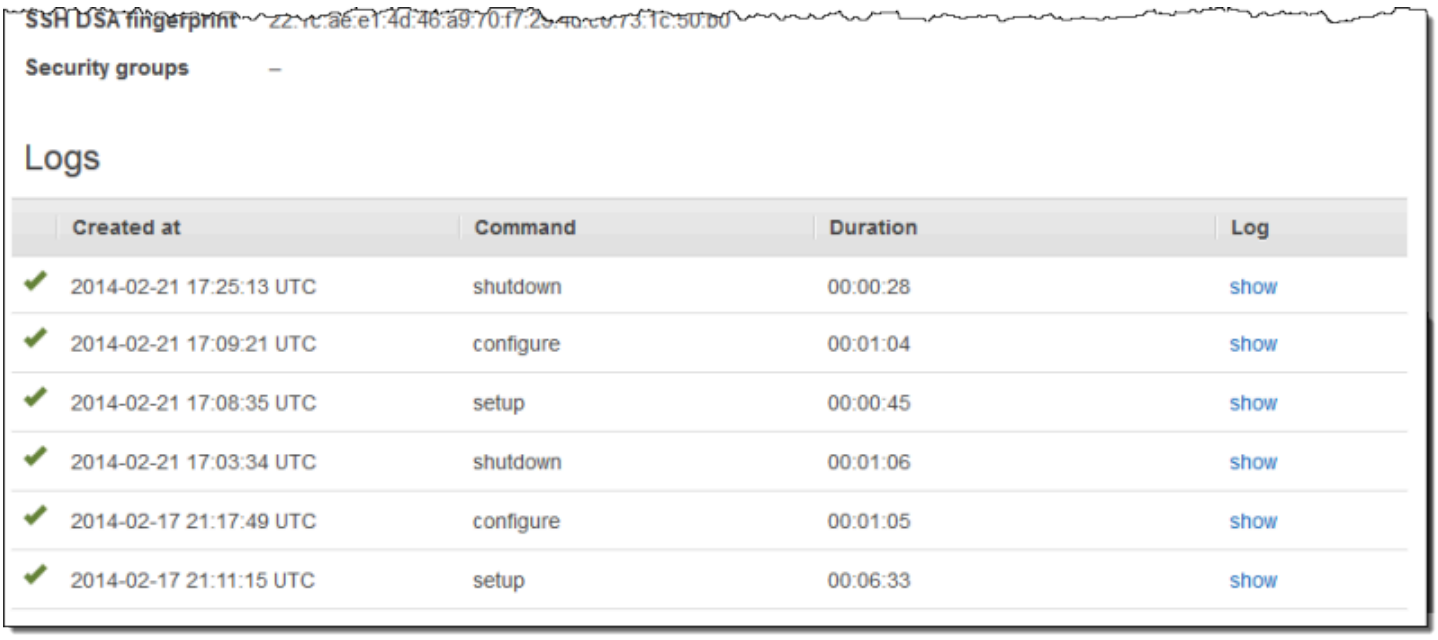
1. AWS OpsWorks Stacks exécute les Setup recettes intégrées de l'instance, suivies de toutes les Setup recettes personnalisées.
2. AWS OpsWorks Stacks exécute les Deploy recettes intégrées de l'instance, suivies de toutes les Deploy recettes personnalisées.

L'instance est désormais en ligne.

3. AWS OpsWorks Stacks déclenche un Configure événement sur toutes les instances de la pile, y compris l'instance nouvellement démarrée.

   AWS OpsWorks Stacks exécute les Configure recettes intégrées des instances, suivies de toutes les recettes personnalisées. Configure

> **Note**
>
> Pour voir les événements du cycle de vie qui ont eu lieu sur une instance particulière, accédez à la page Instances et cliquez sur le nom de l'instance pour ouvrir la page de détails. La liste des événements se trouve dans la section Logs au bas de la page. Vous pouvez cliquer sur show dans la colonne Log afin de rechercher un événement dans le journal de Chef. Il fournit des informations détaillées sur la façon dont l'événement a été géré, notamment les recettes qui ont été exécutées. Pour plus d'informations sur l'interprétation des journaux de Chef, consultez Journaux de Chef.



Pour chaque événement du cycle de vie, AWS OpsWorks Stacks installe un ensemble d'attributs de configuration et de déploiement de la pile sur chaque instance qui contient l'état actuel de la pile et, pour les Deploy événements, des informations sur le déploiement. Les attributs incluent des

informations sur les instances disponibles, leurs adresses IP, etc. Pour de plus amples informations, veuillez consulter Attributs de déploiement et de configuration de pile.

> **Note**
>
> Le démarrage ou l'arrêt d'un grand nombre d'instances en même temps peut rapidement générer un grand nombre d'événements Configure. Pour éviter tout traitement inutile, AWS OpsWorks Stacks ne répond qu'au dernier événement. Les attributs de configuration et de déploiement de la pile de cet événement contiennent toutes les informations requises pour mettre à jour les instances de la pile pour l'ensemble des modifications. Cela vous évite de devoir également traiter les Configure événements antérieurs. AWS OpsWorks Stacks indique que les Configure événements non traités sont remplacés.

Exécution automatique des recettes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Chaque couche a un ensemble de recettes intégrées attribué à chaque événement du cycle de vie, bien que certaines couches manquent de recettes Undeploy. Lorsqu'un événement du cycle de vie se produit sur une instance, AWS OpsWorks Stacks exécute l'ensemble de recettes approprié pour la couche associée.

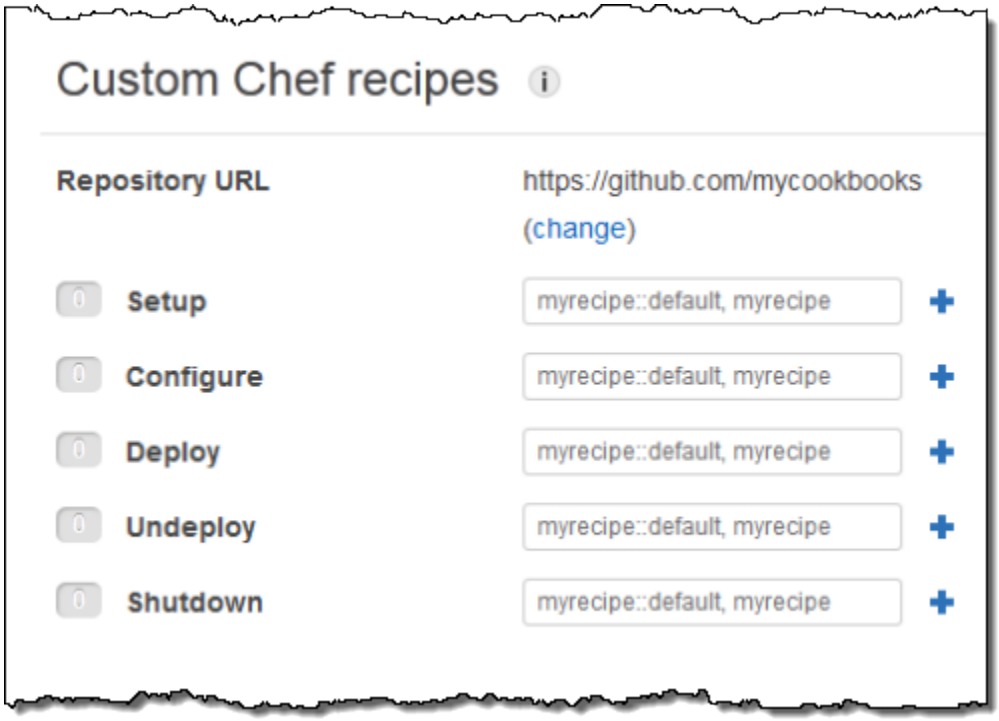
Si vous avez installé des livres de recettes personnalisés, vous pouvez demander à AWS OpsWorks Stacks d'exécuter automatiquement certaines ou toutes les recettes en affectant chaque recette à un événement du cycle de vie d'une couche. Lorsqu'un événement se produit, AWS OpsWorks Stacks exécute les recettes personnalisées spécifiées après les recettes intégrées de la couche.

Pour attribuer des recettes personnalisées aux événements de la couche

1. Sur la page Layers (Couches), pour la couche appropriée, cliquez sur Recipes (Recettes), puis sur Edit (Modifier). Si vous n'avez pas encore activé les livres de recettes personnalisés,

cliquez sur configure cookbooks pour ouvrir la page Settings (Paramètres) de la pile. Basculez Use custom Chef Cookbooks (Utiliser des livres de recettes Chef personnalisés) sur Yes (Oui) et fournissez les informations de référentiel du livre de recettes. Cliquez ensuite sur Save (Enregistrer) et revenez à la page de modification pour l'onglet Recipes (Recettes). Pour de plus amples informations, veuillez consulter Installation de livres de recettes personnalisés.

2. Sous l'onglet Recipes (Recettes), entrez chaque recette personnalisée dans le champ d'événement approprié et cliquez sur + pour l'ajouter à la liste. Spécifiez une recette comme suit : *cookbook* : : *somerecipe* (omettez l'`.rb`extension).



Lorsque vous démarrez une nouvelle instance, AWS OpsWorks Stacks exécute automatiquement les recettes personnalisées pour chaque événement, après avoir exécuté les recettes standard.

> **Note**
>
> Les recettes personnalisées sont exécutées dans l'ordre dans lequel vous les avez entrées dans la console. Un autre moyen de contrôler l'ordre d'exécution consiste à implémenter une recette meta qui exécute les recettes dans l'ordre correct.

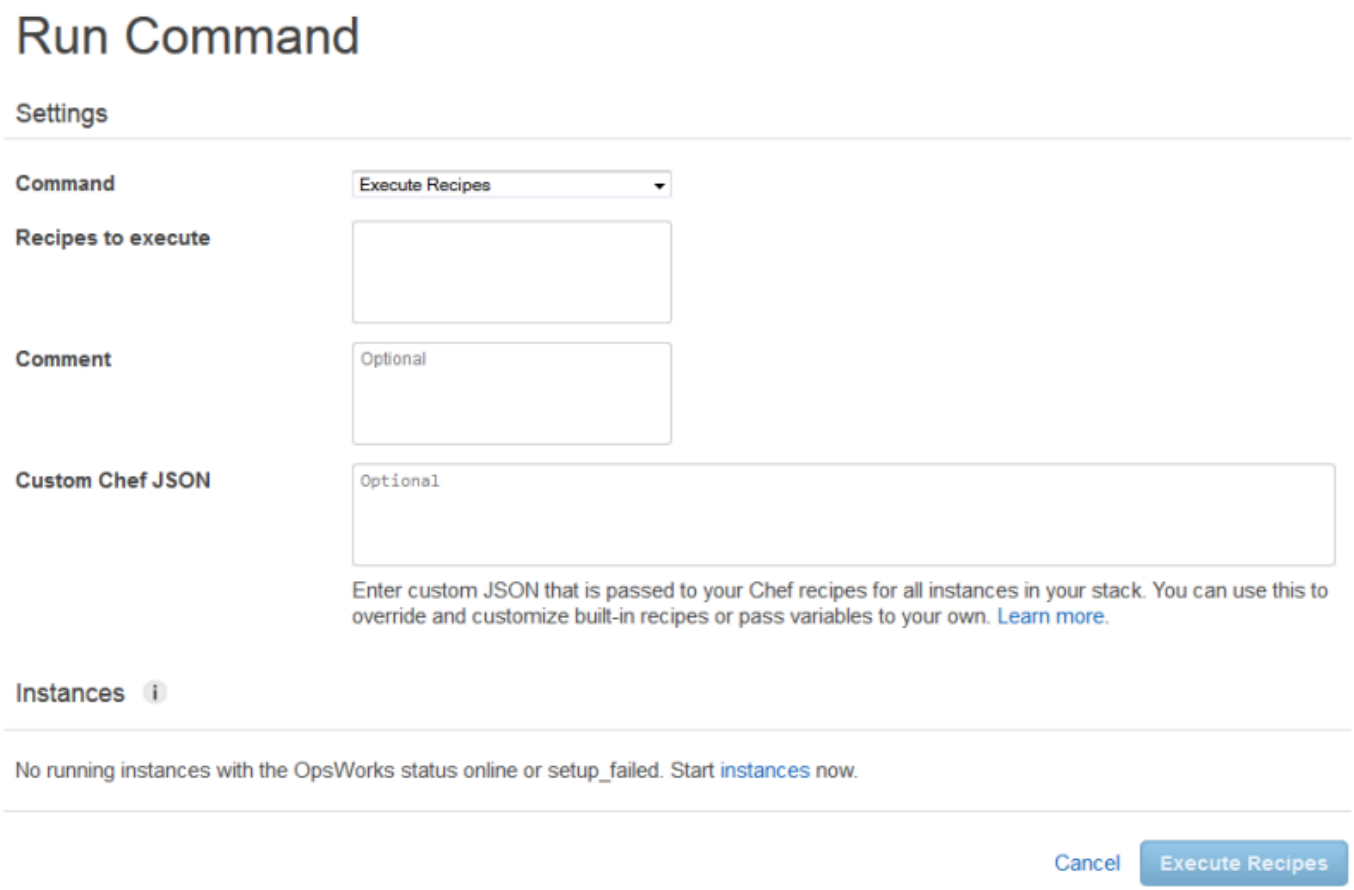
Exécution manuelle des recettes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Bien que les recettes soient généralement exécutées automatiquement en réponse à des événements de cycle de vie, vous pouvez exécuter manuellement des recettes à tout moment sur toutes les instances de la pile ou sur certaines d'entre elles. Cette fonction est généralement utilisée pour les tâches qui ne correspondent pas bien à un événement du cycle de vie, par exemple la sauvegarde des instances. Pour que vous puissiez exécuter une recette personnalisée manuellement, elle doit être dans l'un de vos livres de recettes personnalisés, mais elle ne doit être attribuée à aucun événement du cycle de vie. Lorsque vous exécutez une recette manuellement, AWS OpsWorks Stacks installe les mêmes `deploy` attributs que pour un événement Deploy.

Pour exécuter manuellement des recettes sur les instances de la pile

1. Sur la page Stack (Pile), cliquez sur Run command (Exécuter une commande). Pour Command (Commande), sélectionnez Execute Recipes (Exécuter des recettes).



2. Entrez les recettes à exécuter dans la zone Recettes à exécuter en utilisant le *recipename* format standard *cookbookname* : :. Utilisez des virgules pour séparer plusieurs recettes ; elles s'exécuteront dans l'ordre de la liste.
3. Le cas échéant, utilisez la zone Custom Chef JSON (JSON Chef personnalisé) pour ajouter un objet JSON personnalisé définissant les attributs personnalisés qui seront fusionnés dans les attributs de configuration et de déploiement de la pile qui sont installés sur les instances. Pour plus d'informations sur l'utilisation des objets JSON personnalisés, consultez Utilisation du JSON personnalisé et Remplacement des attributs.
4. Sous Instances, sélectionnez les instances sur lesquelles AWS OpsWorks Stacks doit exécuter les recettes.

Lorsqu'un événement du cycle de vie se produit, l'agent AWS OpsWorks Stacks reçoit une commande pour exécuter les recettes associées. Vous pouvez exécuter ces commandes manuellement sur une instance particulière en utilisant la commande de pile appropriée ou à l'aide de la commande run_command de l'interface de ligne de commande de l'agent. Pour plus

d'informations sur l'utilisation de l'interface de ligne de commande de l'agent, consultez AWS OpsWorks CLI de l'agent Stacks.

Gestion des ressources

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

La page Ressources vous permet d'utiliser l'adresse IP élastique, le volume Amazon EBS ou les ressources d'instance Amazon RDS de votre compte dans une pile AWS OpsWorks Stacks. Vous pouvez utiliser Ressources pour effectuer les tâches suivantes :

- Enregistrer une ressource auprès d'une pile, ce qui vous permet d'attacher la ressource à l'une des instances de la pile.
- Attacher une ressource à l'une des instances de la pile.
- Déplacer une ressource d'une instance à l'autre.
- Détacher une ressource d'une instance. La ressource demeure enregistrée et peut être attachée à une autre instance.
- Annuler l'enregistrement d'une ressource. Une ressource non enregistrée ne peut pas être utilisée par AWS OpsWorks Stacks, mais elle reste dans votre compte à moins que vous ne la supprimiez, et elle peut être enregistrée auprès d'une autre pile.

Notez les contraintes suivantes :

- Vous ne pouvez pas associer de volumes Amazon EBS enregistrés à des instances Windows.
- La page Ressources gère les volumes Amazon EBS standard, PIOPS, à débit optimisé, à froid ou à usage général (SSD) Amazon EBS, mais pas les matrices RAID.
- Les volumes Amazon EBS doivent être au format xfs.

AWS OpsWorks Stacks ne prend pas en charge les autres formats de fichiers, tels que ext4. Pour plus d'informations sur la préparation des volumes Amazon EBS, consultez la section Mise en service d'un volume Amazon EBS.

- Vous ne pouvez pas associer un volume Amazon EBS à une instance en cours d'exécution, ni le détacher d'une telle instance.

  Vous pouvez opérer uniquement sur les instances hors connexion. Par exemple, vous pouvez enregistrer un volume en cours d'utilisation auprès d'une pile et l'attacher à une instance hors connexion, mais vous devez arrêter l'instance originale et détacher le volume avant de commencer la nouvelle instance. Sinon, le processus de démarrage échoue.
- Toutes les ressources enregistrées sont gérées uniquement dans AWS OpsWorks. Cela peut remplacer les propriétés du cycle de vie des ressources, par exemple `DeleteOnTermination` pour les EC2 volumes.
- Vous pouvez attacher une adresse IP Elastic à une instance en cours d'exécution ou l'en détacher.

  Vous pouvez opérer sur les instances en ligne ou hors connexion. Par exemple, vous pouvez enregistrer une adresse en cours d'utilisation et l'attribuer à une instance en cours d'exécution, et AWS OpsWorks Stacks réattribuera automatiquement l'adresse.
- Pour enregistrer et annuler l'enregistrement des ressources, votre stratégie IAM doit accorder des autorisations pour les actions suivantes :

Volumes Amazon EBS	Adresses IP Elastic	Instances Amazon RDS
RegisterVolume	RegisterElasticIp	RegisterRdsDbInstance
UpdateVolume	UpdateElasticIp	UpdateRdsDbInstance
DeregisterVolume	DeregisterElasticIp	DeregisterRdsDbInstance

Le niveau Gérer les autorisations accorde les autorisations pour toutes ces actions. Pour empêcher un utilisateur Manage d'enregistrer des ressources particulières ou d'annuler leur enregistrement, modifiez leur stratégie IAM pour refuser les autorisations des actions appropriées. Pour de plus amples informations, veuillez consulter Sécurité et autorisations.

Rubriques

- Enregistrement des ressources auprès d'une pile
- Attachement et déplacement de ressources
- Détachement des ressources
- Annulation de l'enregistrement des ressources

Enregistrement des ressources auprès d'une pile

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les volumes Amazon EBS ou les adresses IP Elastic doivent être enregistrés auprès d'une pile avant de pouvoir les associer à des instances. Lorsque AWS OpsWorks Stacks crée des ressources pour une pile, elles sont automatiquement enregistrées auprès de cette pile. Si vous souhaitez utiliser des ressources créées de façon externe, vous devez les enregistrer explicitement. Remarques :

- Vous pouvez enregistrer une ressource auprès d'une seule pile à la fois.
- Lorsque vous supprimez une pile, AWS OpsWorks Stacks annule l'enregistrement de toutes les ressources.

Rubriques

- Enregistrement de volumes Amazon EBS auprès d'une pile
- Enregistrement des adresses IP Elastic auprès d'une pile
- Enregistrement d'instances Amazon RDS auprès d'une pile

Enregistrement de volumes Amazon EBS auprès d'une pile

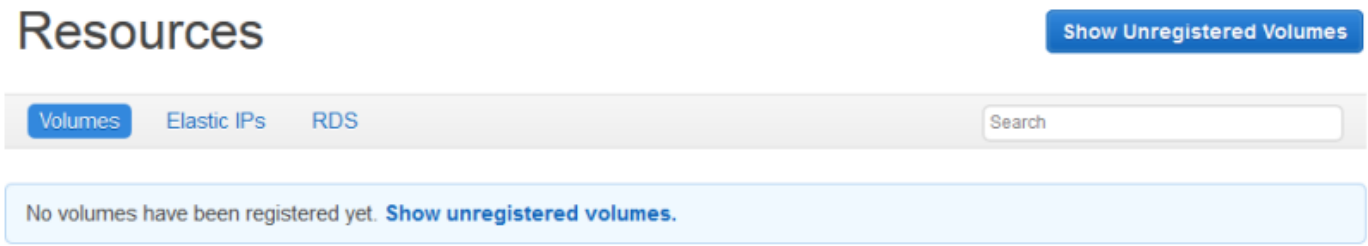
> **Note**
>
> Cette ressource ne peut être utilisée qu'avec les piles Linux. Bien que vous puissiez enregistrer un volume Amazon EBS auprès d'une pile Windows, vous ne pouvez pas l'associer à une instance.

Vous pouvez utiliser la page Ressources pour enregistrer un volume Amazon EBS auprès d'une pile, sous réserve des contraintes suivantes :

- Les volumes Amazon EBS attachés et non root doivent être des disques durs standard, à débit optimisé, à froid, PIOPS ou à usage général (SSD), mais pas une matrice RAID. Pour plus d'informations sur les tailles maximale et minimale du volume, consultez Volumes EBS dans ce guide.
- Les volumes doivent être au format XFS.
- AWS OpsWorks Stacks ne prend pas en charge les autres formats de fichiers, tels que le quatrième système de fichiers étendu (ext4), pour les volumes Amazon EBS non root. Pour plus d'informations sur la préparation des volumes Amazon EBS, consultez la section Mise en service d'un volume Amazon EBS. Notez que l'exemple de cette rubrique décrit comment créer un volume ext4, mais vous pouvez suivre les mêmes procédures pour les volumes XFS.

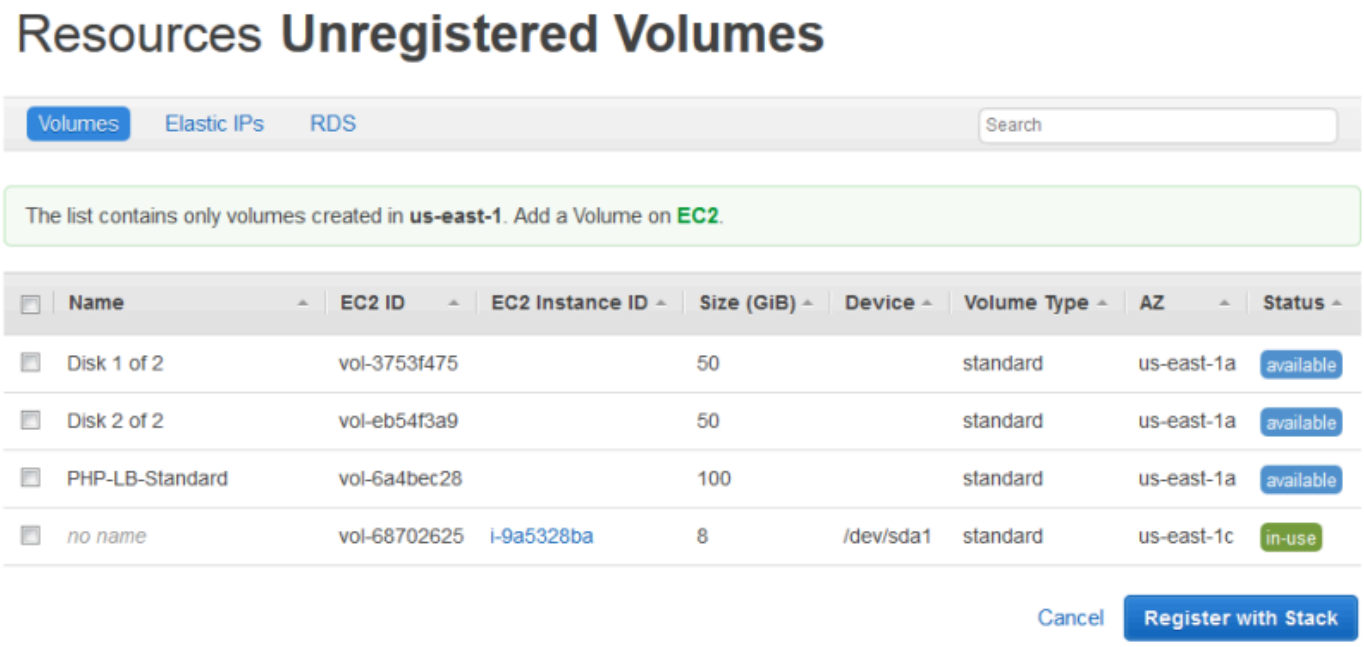
Pour enregistrer un volume Amazon EBS

1. Ouvrez la pile souhaitée et cliquez sur Resources dans le volet de navigation.
2. Cliquez sur Volumes pour afficher les volumes Amazon EBS disponibles. A l'origine, la pile n'a pas de volumes enregistrés, comme illustré ci-après.

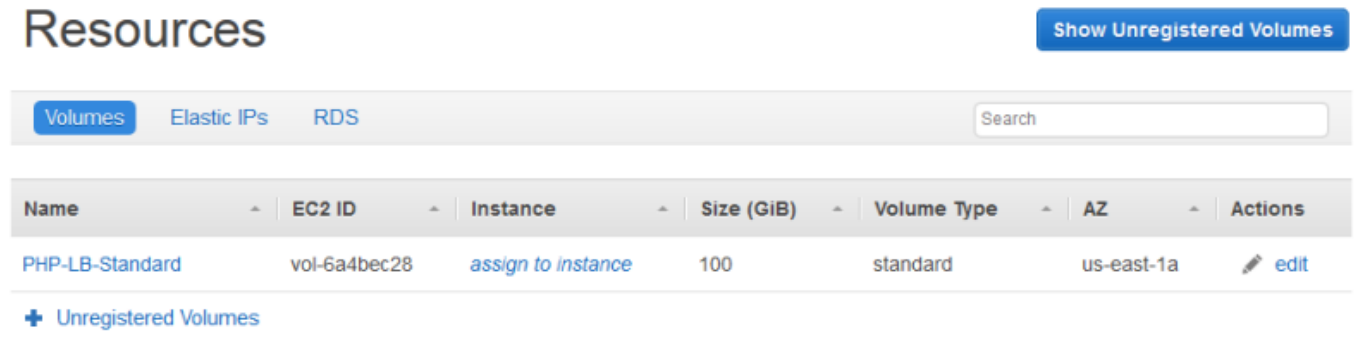


3. Cliquez sur Afficher les volumes non enregistrés pour afficher les volumes Amazon EBS de votre compte qui se trouvent dans la région de la pile et, le cas échéant, le VPC de la pile. La colonne Statut indique si les volumes sont disponibles en vue de leur utilisation. Type de volume indique

si le volume est standard (`standard`), SSD à usage général (gp2), PIOPS (`io1`, suivi de la valeur IOPS par disque entre parenthèses), HDD à débit optimisé (`st1`) ou HDD à froid (`sc1`).



4. Sélectionnez les volumes appropriés et cliquez sur Register to Stack (Enregistrer sur la pile). La page Ressources affiche désormais les volumes nouvellement enregistrés.



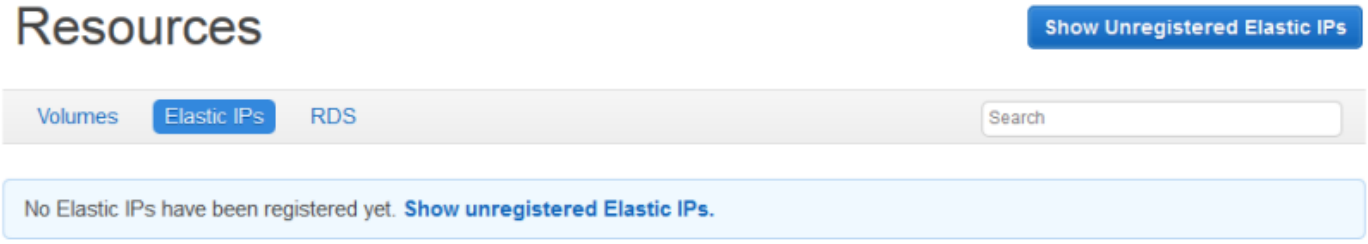
Pour enregistrer des volumes supplémentaires, cliquez sur Show Unregistered Volumes (Afficher les volumes non enregistrés) ou + Unregistered Volumes (+ de volumes non enregistrés), puis répétez l'opération.

Enregistrement des adresses IP Elastic auprès d'une pile

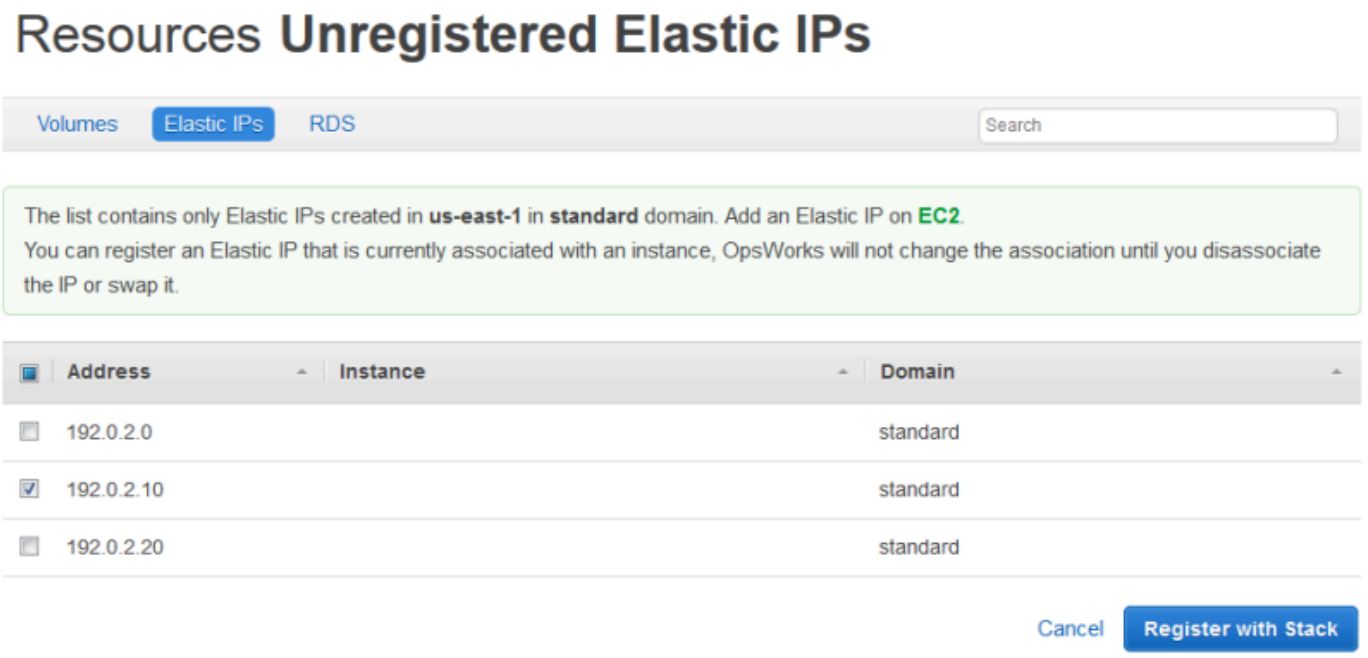
Utilisez la procédure suivante pour enregistrer les adresses IP Elastic.

Pour enregistrer une adresse IP Elastic.

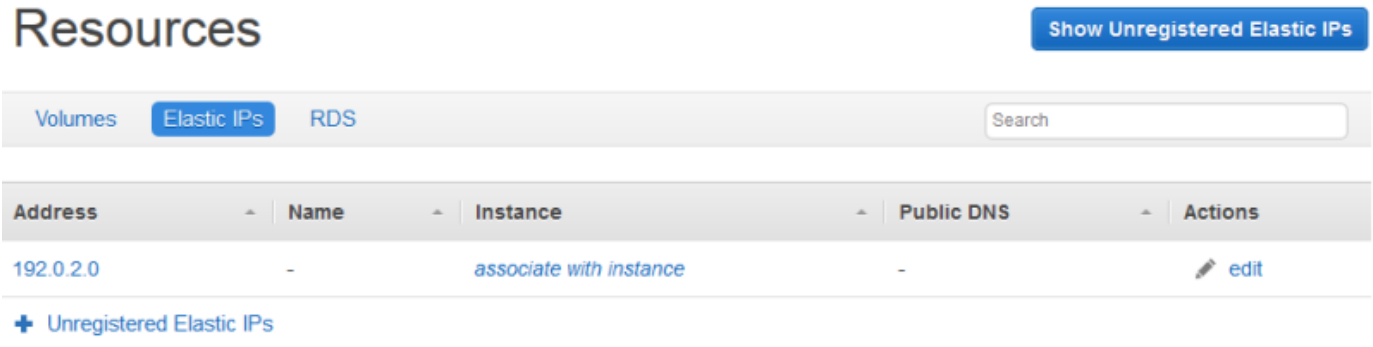
1. Ouvrez la page Ressources de la pile et cliquez sur Elastic IPs pour afficher les adresses IP Elastic disponibles. A l'origine, la pile n'a pas d'adresses enregistrées, comme illustré ci-après.



2. Cliquez sur Afficher les adresses IP élastiques non enregistrées IPs pour afficher les adresses IP élastiques disponibles dans votre compte qui se trouvent dans la région de la pile.



3. Sélectionnez les adresses appropriées et cliquez sur Register to Stack (Enregistrer sur la pile). Vous retournez alors à la page Ressources, qui affiche maintenant les adresses nouvellement enregistrées.



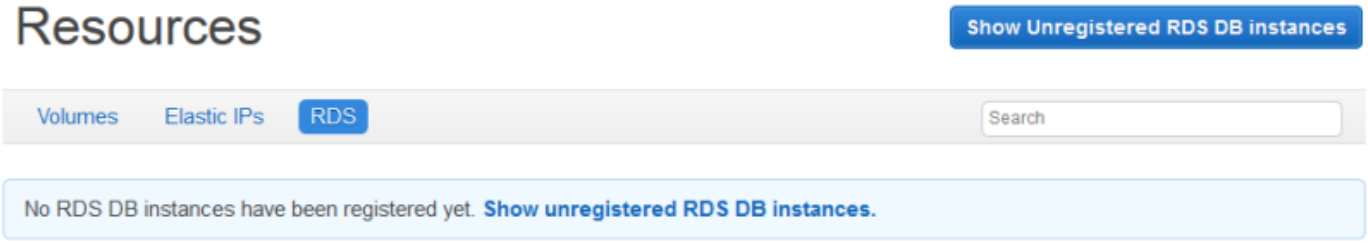
Pour enregistrer des adresses supplémentaires, cliquez sur Afficher l'élastique non enregistré IPs ou sur + Elastic non enregistré IPs et répétez cette procédure.

Enregistrement d'instances Amazon RDS auprès d'une pile

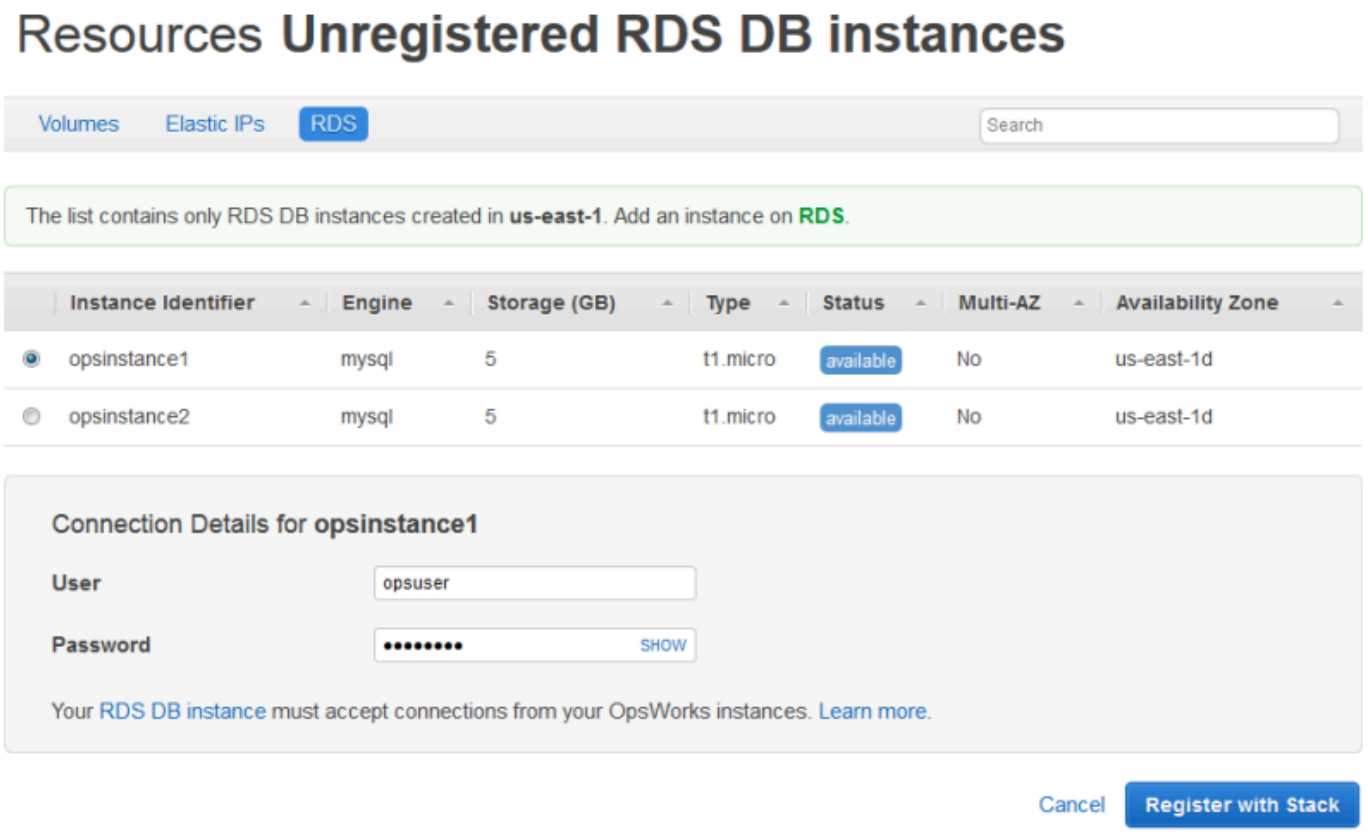
Utilisez la procédure suivante pour enregistrer des instances Amazon RDS.

Pour enregistrer une instance Amazon RDS

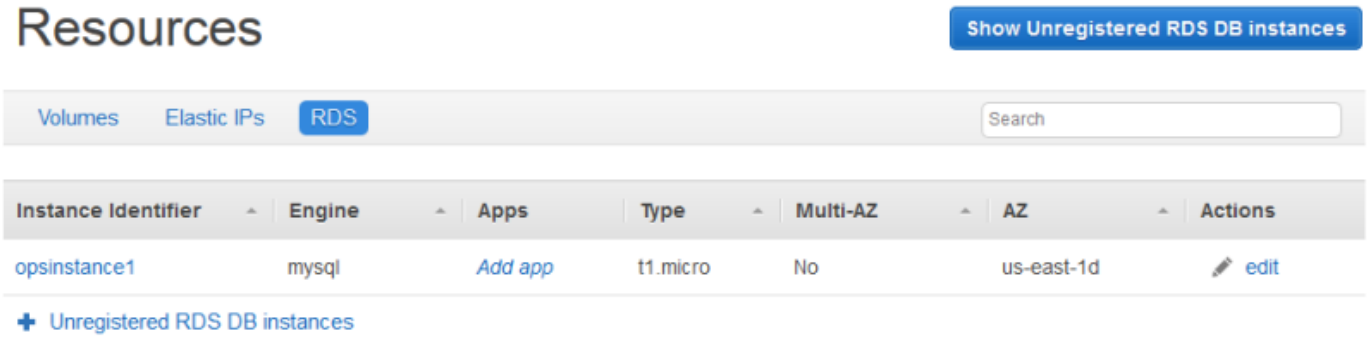
1. Ouvrez la page Ressources de la pile et cliquez sur RDS pour afficher les instances Amazon RDS disponibles. A l'origine, la pile n'a pas d'instances enregistrées, comme illustré ci-après.



2. Cliquez sur Afficher les instances de base de données RDS non enregistrées pour afficher les instances Amazon RDS disponibles sur votre compte qui se trouvent dans la région de la pile.



3. Sélectionnez l'instance appropriée, entrez ses valeurs utilisateur principal et mot de passe principal pour Utilisateur et Mot de passe, puis cliquez sur Register to Stack (Enregistrer sur la pile). Vous retournez alors à la page Ressources, qui affiche maintenant l'instance nouvellement enregistrée.



> **Important**
>
> Vous devez vous assurer que l'utilisateur et le mot de passe que vous utilisez pour enregistrer l'instance Amazon RDS correspondent à un utilisateur et à un mot de passe valides. Dans le cas contraire, vos applications ne pourront pas se connecter à l'instance.

Pour enregistrer des adresses supplémentaires, cliquez sur Show Unregistered RDS DB instances ou + Unregistered RDS DB instances, puis répétez l'opération. Pour plus d'informations sur l'utilisation des instances Amazon RDS avec AWS OpsWorks Stacks, consultez. Couche de service Amazon RDS

> **Note**
>
> Vous pouvez également enregistrer des instances Amazon RDS via la page Layers. Pour de plus amples informations, veuillez consulter Couche de service Amazon RDS.

Attachement et déplacement de ressources

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une fois que vous avez enregistré une ressource auprès d'une pile, vous pouvez l'attacher à l'une des instances de la pile. Vous pouvez également déplacer une ressource attachée depuis une instance vers une autre. Remarques :

- Lorsque vous attachez ou déplacez des volumes Amazon EBS, les instances impliquées dans l'opération doivent être hors ligne. Si l'instance qui vous intéresse n'est pas sur la page Ressources, accédez à la page Instances et arrêtez l'instance. Une fois qu'elle a été arrêtée, vous pouvez revenir à la page Ressources, puis attacher ou déplacer la ressource.
- Lorsque vous attachez ou déplacez les adresses IP Elastic, les instances peuvent être en ligne ou hors connexion.
- Si vous supprimez une instance, toutes les ressources attachées demeurent enregistrées auprès de la pile. Vous pouvez alors attacher la ressource à une autre instance ou, si vous n'en avez plus besoin, annuler l'enregistrement de la ressource.

Rubriques

- Affectation de volumes Amazon EBS à une instance
- Association des adresses IP Elastic auprès d'une instance
- Associer des instances Amazon RDS à une application

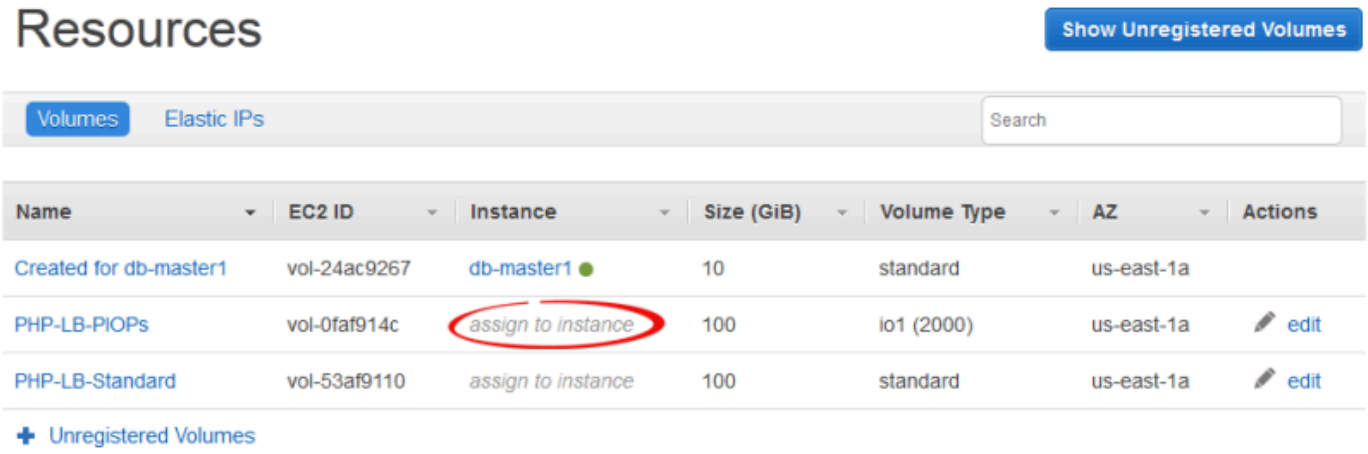
Affectation de volumes Amazon EBS à une instance

> **Note**
>
> Vous ne pouvez pas attribuer de volumes Amazon EBS à des instances Windows.

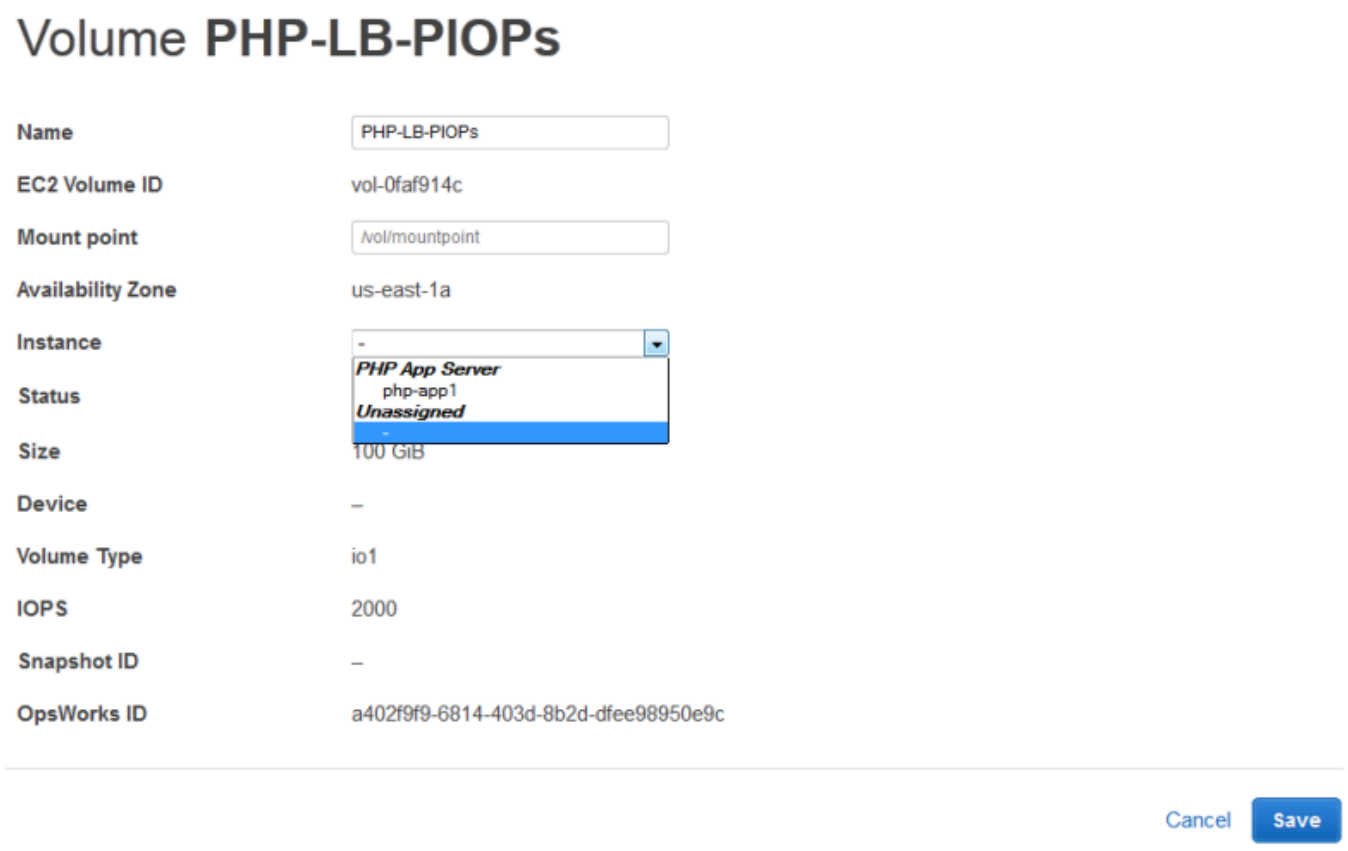
Vous pouvez attribuer un volume Amazon EBS enregistré à une instance et le déplacer d'une instance à l'autre, mais les deux instances doivent être hors ligne.

Pour attribuer un volume Amazon EBS à une instance

1. Sur la page Ressources, cliquez sur Attribuer à une instance dans la colonne Instance du volume approprié.



2. Sur la page des détails du volume, sélectionnez l'instance appropriée, spécifiez le nom du volume et le point de montage, puis cliquez sur Enregistrer pour attacher le volume à l'instance.



> **Important**
>
> Si vous avez attribué un volume externe en cours d'utilisation à votre instance, vous devez utiliser la EC2 console, l'API ou la CLI Amazon pour le supprimer de l'instance d'origine, sinon le processus de démarrage échouera.

Vous pouvez également utiliser la page de détails pour déplacer un volume Amazon EBS attribué vers une autre instance de la pile.

Pour déplacer un volume Amazon EBS vers une autre instance

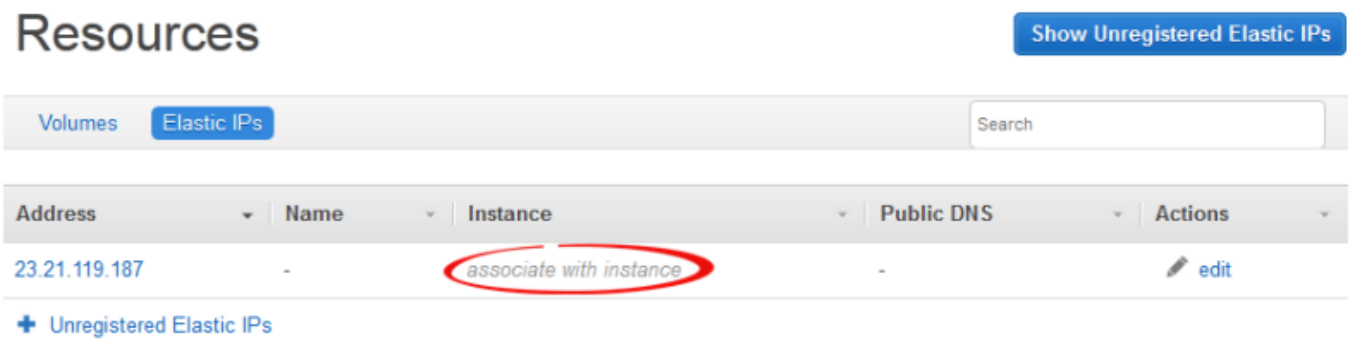
1. Assurez-vous que les deux instances se trouvent dans l'état hors connexion.
2. Sur la page Ressources, cliquez sur Volumes, puis cliquez sur Modifier dans la colonne Actions du volume.
3. Effectuez l'une des actions suivantes :
   - Pour déplacer le volume vers une autre instance de la pile, sélectionnez l'instance appropriée dans la liste Instance, puis cliquez sur Enregistrer.
   - Pour déplacer le volume vers une instance d'une autre pile, annulez l'enregistrement du volume, enregistrez le volume auprès de la nouvelle pile et attachez-le à la nouvelle instance.

Association des adresses IP Elastic auprès d'une instance

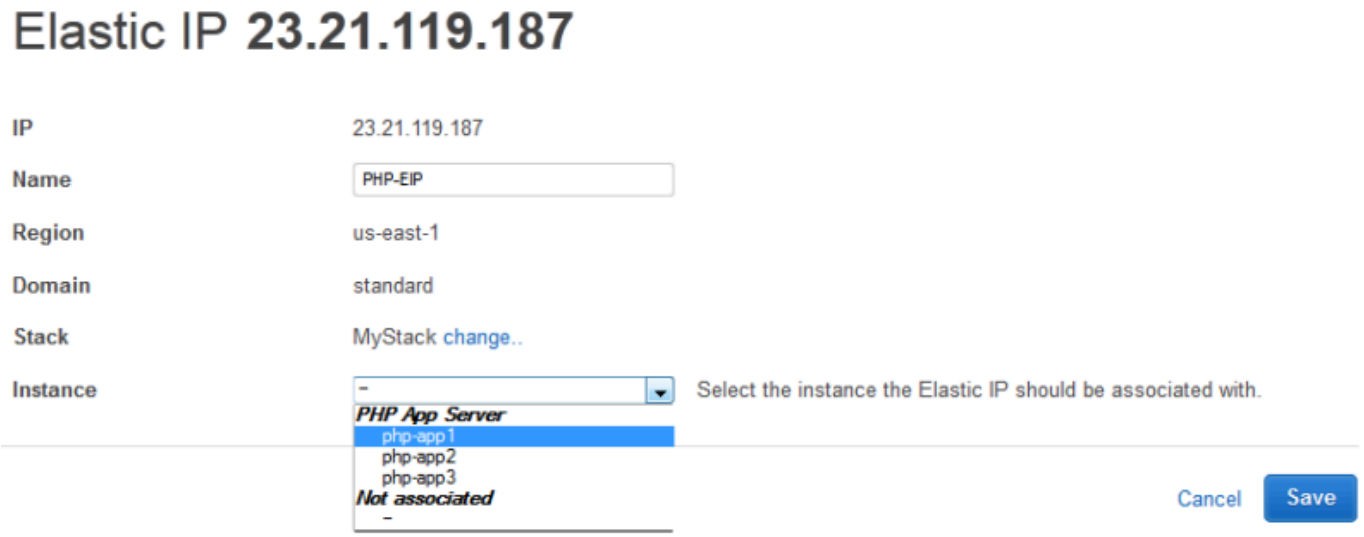
Vous pouvez associer une adresse IP Elastic enregistrée auprès d'une instance et la transférer d'une instance vers une autre, y compris les instances d'autres piles. Les instances peuvent être en ligne ou hors connexion.

Pour associer une adresse IP Elastic à une instance

1. Sur la page Ressources, cliquez sur Associer à l'instance dans la colonne Instance de l'adresse appropriée.



2. Sur la page des détails de l'adresse, sélectionnez l'instance appropriée, spécifiez le nom de l'adresse et cliquez sur Enregistrer pour associer l'adresse à l'instance.



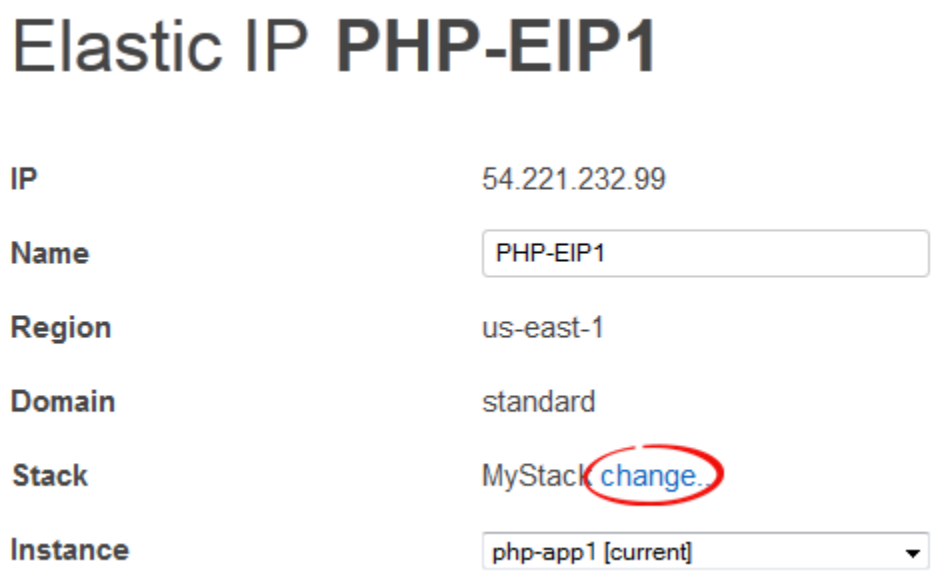
> **Note**
>
> Si l'adresse IP Elastic est actuellement associée à une autre instance en ligne, AWS OpsWorks Stacks la réaffecte automatiquement à la nouvelle instance.

Vous pouvez aussi utiliser la page des détails pour déplacer une adresse IP Elastic associée vers une autre instance.

Pour déplacer une adresse IP Elastic vers une autre instance

1. Sur la page Ressources, cliquez sur Elastic, IPs puis sur Modifier dans la colonne Actions de l'adresse.
2. Effectuez l'une des actions suivantes :

- Pour déplacer l'adresse vers une autre instance de la pile, sélectionnez l'instance appropriée dans la liste Instance, puis cliquez sur Enregistrer.
- Pour déplacer l'adresse vers une instance d'une autre pile, cliquez sur Modifier dans les paramètres Pile pour afficher la liste des piles disponibles. Sélectionnez une pile à partir de la liste Pile et une instance à partir de la liste Instance. Puis, cliquez sur Save (Enregistrer).



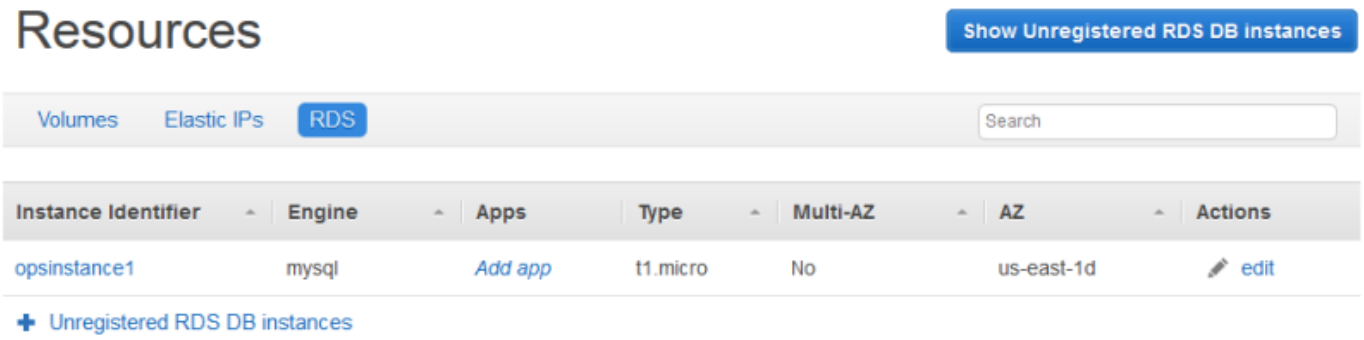
Une fois que vous avez joint ou déplacé une adresse, AWS OpsWorks Stacks déclenche un événement de configuration du cycle de vie pour informer les instances de la pile de la modification.

Associer des instances Amazon RDS à une application

Vous pouvez associer une instance Amazon RDS à une ou plusieurs applications.

Pour associer une instance Amazon RDS à une application

1. Sur la page Ressources, cliquez sur Ajouter une application dans la colonne Applications de l'instance appropriée.



2. Utilisez la page Ajouter une application pour joindre l'instance Amazon RDS. Pour de plus amples informations, veuillez consulter Ajout d'applications.

Étant donné qu'un Amazon RDS peut être attaché à plusieurs applications, il n'existe aucune procédure spéciale pour déplacer l'instance d'une application à l'autre. Modifiez simplement la première application pour supprimer l'instance RDS ou modifiez la deuxième application pour ajouter l'instance RDS. Pour de plus amples informations, veuillez consulter Modification des applications.

Détachement des ressources

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Lorsque vous n'avez plus besoin d'une ressource attachée, vous pouvez la détacher. Cette ressource reste enregistrée auprès de la pile et peut être attachée ailleurs.

Rubriques

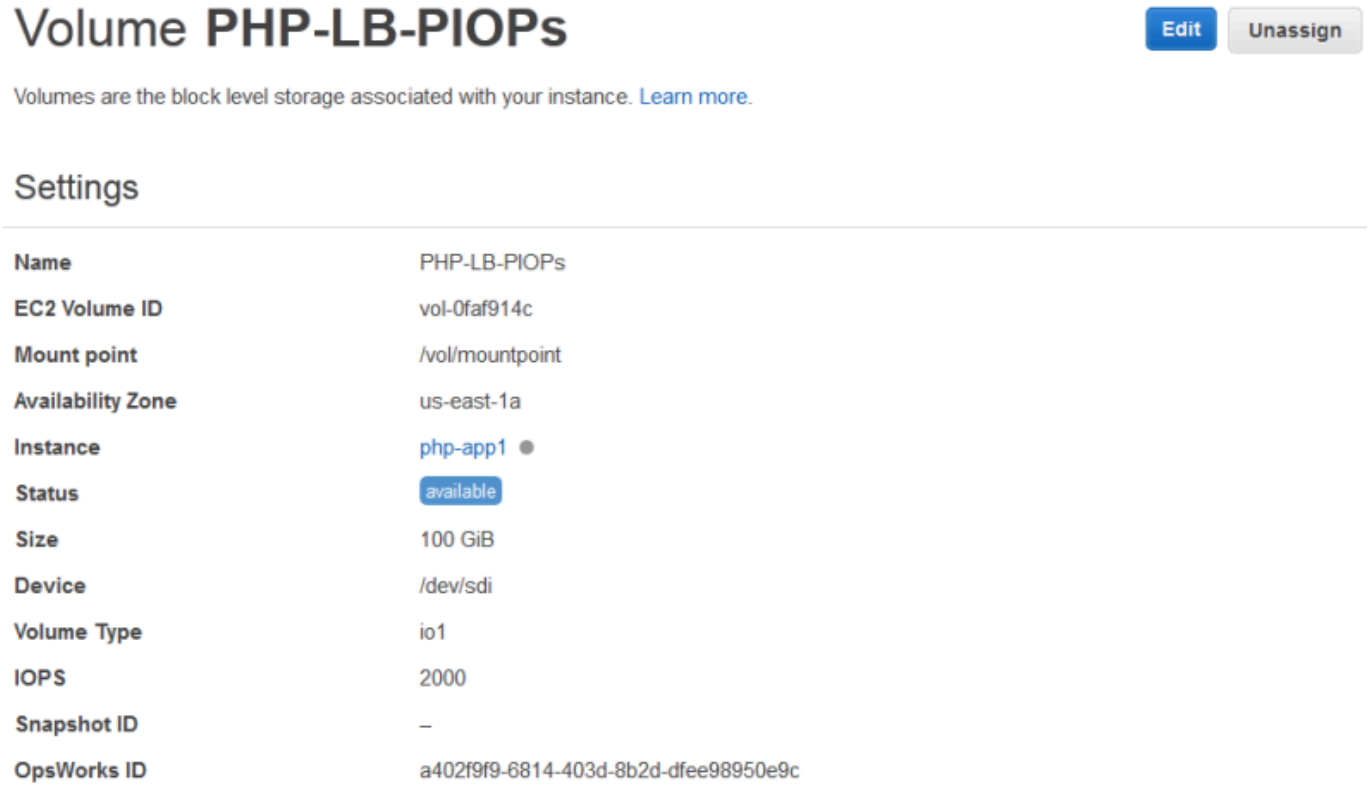
- Annulation de l'attribution de volumes Amazon EBS
- Dissociation d'adresses IP Elastic
- Détachement d'instances Amazon RDS

Annulation de l'attribution de volumes Amazon EBS

Utilisez la procédure suivante pour annuler l'attribution d'un volume Amazon EBS à son instance.

Pour annuler l'attribution d'un volume Amazon EBS

1. Assurez-vous que l'instance se trouve dans l'état hors connexion.
2. Sur la page Ressources, cliquez sur Volumes et cliquez sur le nom du volume.
3. Sur la page des détails du volume, cliquez sur Annuler l'attribution.

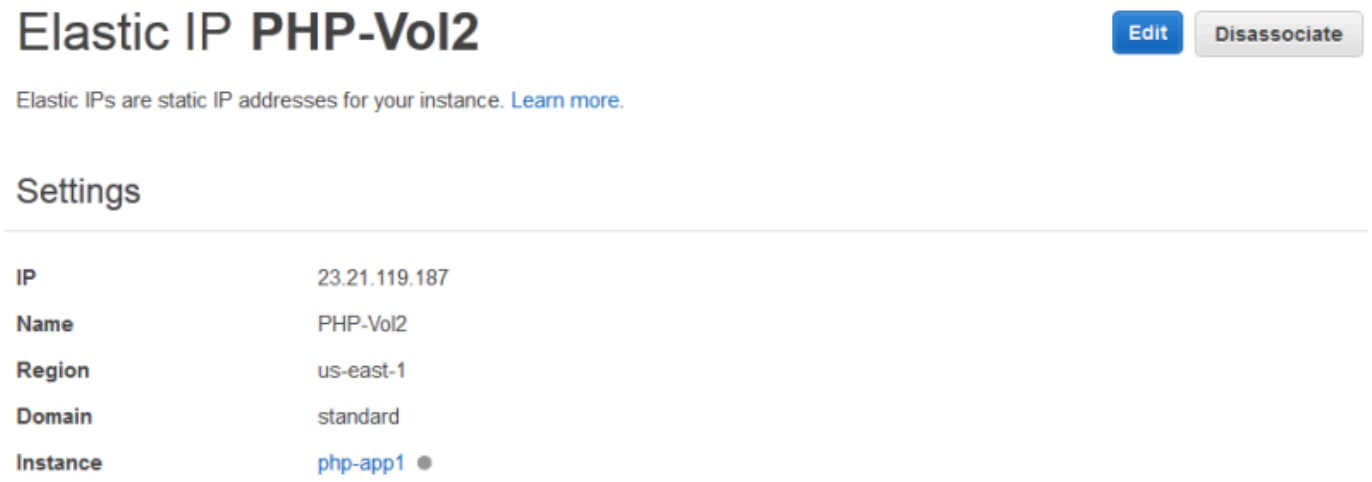


Dissociation d'adresses IP Elastic

Utilisez la procédure suivante pour dissocier une adresse IP Elastic de son instance.

Dissocier une adresse IP Elastic

1. Sur la page Ressources, cliquez sur Elastic, IPs puis sur Modifier dans la colonne Actions de l'adresse.
2. Sur la page des détails de l'adresse, cliquez sur Dissocier.



Une fois que vous avez dissocié une adresse, AWS OpsWorks Stacks déclenche un événement de configuration du cycle de vie pour informer les instances de la pile de la modification.

Détachement d'instances Amazon RDS

Utilisez la procédure suivante pour détacher un Amazon RDS d'une application.

Pour détacher une instance Amazon RDS

1. Sur la page Ressources, cliquez sur RDS et cliquez sur l'application appropriée dans la colonne Applications.
2. Cliquez sur Modifier et modifiez la configuration de l'application pour détacher l'instance. Pour de plus amples informations, veuillez consulter Modification des applications.

> **Note**
>
> Cette procédure détache un Amazon RDS d'une seule application. Si l'instance est attachée à plusieurs applications, vous devez répéter cette procédure pour chaque application.

Annulation de l'enregistrement des ressources

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement

aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Si vous n'avez plus besoin d'avoir une ressource enregistrée auprès d'une pile, vous pouvez en annuler l'enregistrement. La désinscription ne supprime pas la ressource de votre compte ; elle y reste et peut être enregistrée dans une autre pile ou utilisée en dehors de Stacks. AWS OpsWorks Si vous souhaitez supprimer la ressource entièrement, vous avez deux options :

- Si une adresse IP élastique ou une ressource Amazon EBS est attachée à une instance, vous pouvez supprimer la ressource lorsque vous supprimez l'instance.

  Accédez à la page Instances, cliquez sur delete (supprimer) dans la colonne Actions de l'instance, puis sélectionnez Supprimer les volumes EBS de l'instance ou Delete the instance's Elastic IP (Supprimer l'adresse IP Elastic de l'instance).
- Désenregistrez la ressource, puis utilisez la console, l'API ou la CLI EC2 Amazon ou Amazon RDS pour la supprimer.

Rubriques

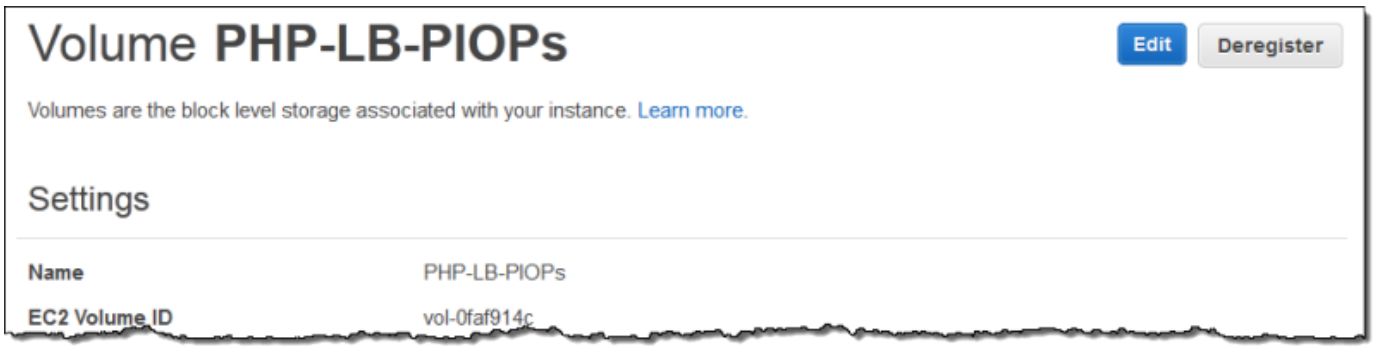
- Annulation de l'enregistrement d'Amazon EBS Volumes
- Annulation de l'enregistrement des adresses IP Elastic
- Annulation de l'enregistrement des instances Amazon RDS

Annulation de l'enregistrement d'Amazon EBS Volumes

Suivez la procédure ci-dessous pour désenregistrer un volume Amazon EBS.

Pour désenregistrer un volume Amazon EBS

1. Si le volume est attaché à une instance, annulez l'attribution, comme décrit dans Annulation de l'attribution de volumes Amazon EBS.
2. Sur la page Ressources, cliquez sur le nom du volume dans la colonne Nom.
3. Sur la page des détails du volume, cliquez sur Annuler l'inscription.

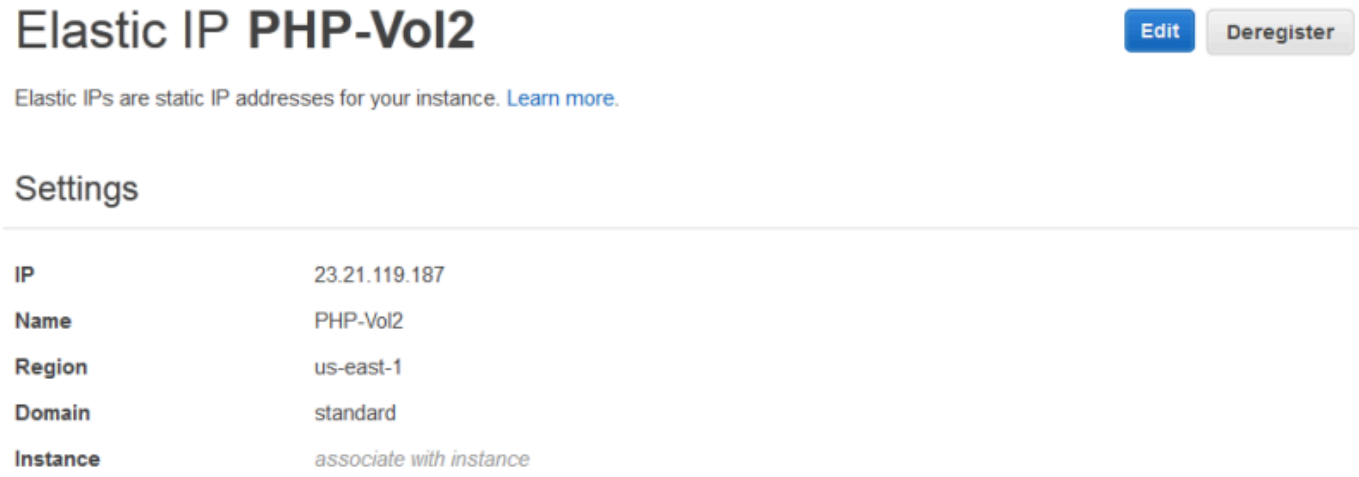


Annulation de l'enregistrement des adresses IP Elastic

Utilisez la procédure suivante pour annuler l'enregistrement d'une adresse IP Elastic.

Pour annuler l'enregistrement d'une adresse IP Elastic

1. Si l'adresse est associée à une instance, dissociez-la, comme décrit dans Dissociation d'adresses IP Elastic.
2. Sur la page Ressources, cliquez sur Elastic, IPs puis sur l'adresse IP dans la colonne Adresse.
3. Sur la page des détails de l'adresse, cliquez sur Annuler l'inscription.



> **Note**
>
> Si vous souhaitez simplement enregistrer une adresse IP Elastic auprès d'une autre pile, vous devez annuler son enregistrement de la pile actuelle, puis l'enregistrer auprès de la nouvelle pile. Cependant, vous pouvez déplacer directement une adresse IP Elastic attachée

vers une instance d'une autre pile. Pour de plus amples informations, veuillez consulter Attachement et déplacement de ressources.

Annulation de l'enregistrement des instances Amazon RDS

Utilisez la procédure suivante pour annuler l'enregistrement d'une instance Amazon RDS.

Pour annuler l'enregistrement d'une instance Amazon RDS

1. Si l'instance est associée à une application, détachez-la, comme décrit dans Détachement des ressources.
2. Sur la page Ressources, cliquez sur RDS, puis sur le nom de l'instance.
3. Sur la page des détails de l'instance, cliquez sur Annuler l'inscription.



Balises

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les balises peuvent vous aider à regrouper des ressources dans des piles Chef 11.10, Chef 12 et Chef 12.2, et suivre les coûts d'utilisation de ces ressources dans AWS Billing and Cost Management.

Vous pouvez appliquer des balises au niveau de la pile et de la couche. Lorsque vous créez une balise, vous appliquez la balise à chaque ressource au sein de la structure balisée. Par exemple, si vous appliquez une balise à une couche, vous l'appliquez à chaque instance, volume Amazon EBS (sauf la racine) ou équilibreur de charge Elastic Load Balancing de la couche. Les balises ne peuvent pas actuellement être appliquées au volume racine, ou volume EBS par défaut, d'une instance.

Les balises sont des paires clé-valeur que vous attribuez à des piles ou à des couches dans Stacks. AWS OpsWorks Après avoir créé des balises, ouvrez la console Billing and Cost Management pour activer les balises définies par l'utilisateur. Pour plus d'informations sur la façon d'activer vos tags et de les utiliser pour suivre et gérer les coûts de vos ressources AWS OpsWorks Stacks, consultez les sections Utilisation des balises de répartition des coûts et activation des balises de répartition des coûts définies par l'utilisateur dans le guide de l'utilisateur de Billing and Cost Management.

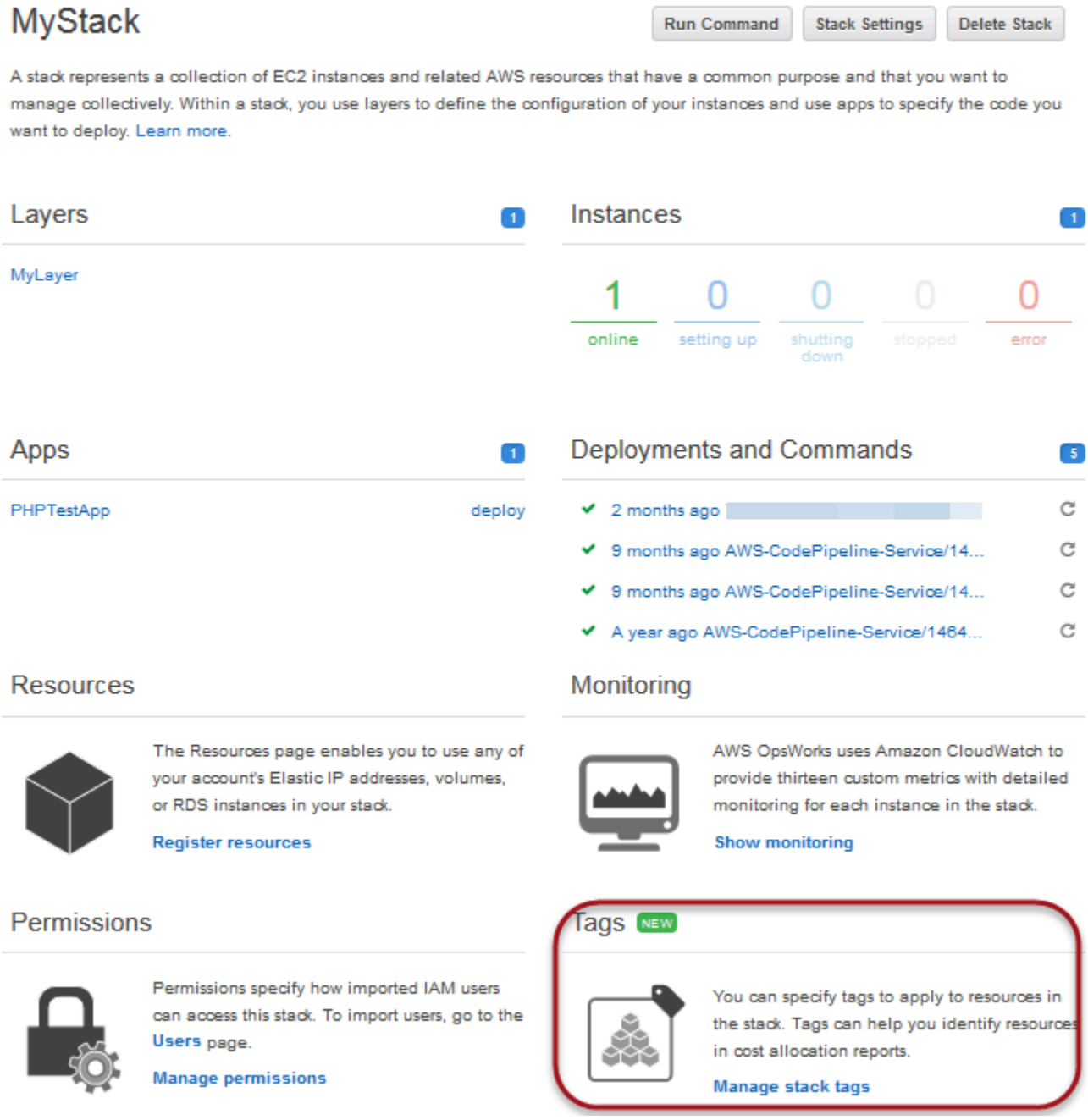
Les balises fonctionnent de la même manière que les attributs personnalisés dans AWS OpsWorks Stacks. Les balises que vous appliquez à une pile sont héritées par chaque couche de la pile. Au niveau de la couche, vous pouvez remplacer les valeurs (mais pas les noms de clé) des balises héritées et ajouter de nouvelles balises spécifiques à la couche. AWS OpsWorks applique le jeu de balises obtenu à toutes les ressources de la couche. A mesure que vous créez de nouvelles ressources ou attribuez des ressources existantes à une couche, les nouvelles ressources de la couche sont balisées avec le même ensemble de balises.

Rubriques

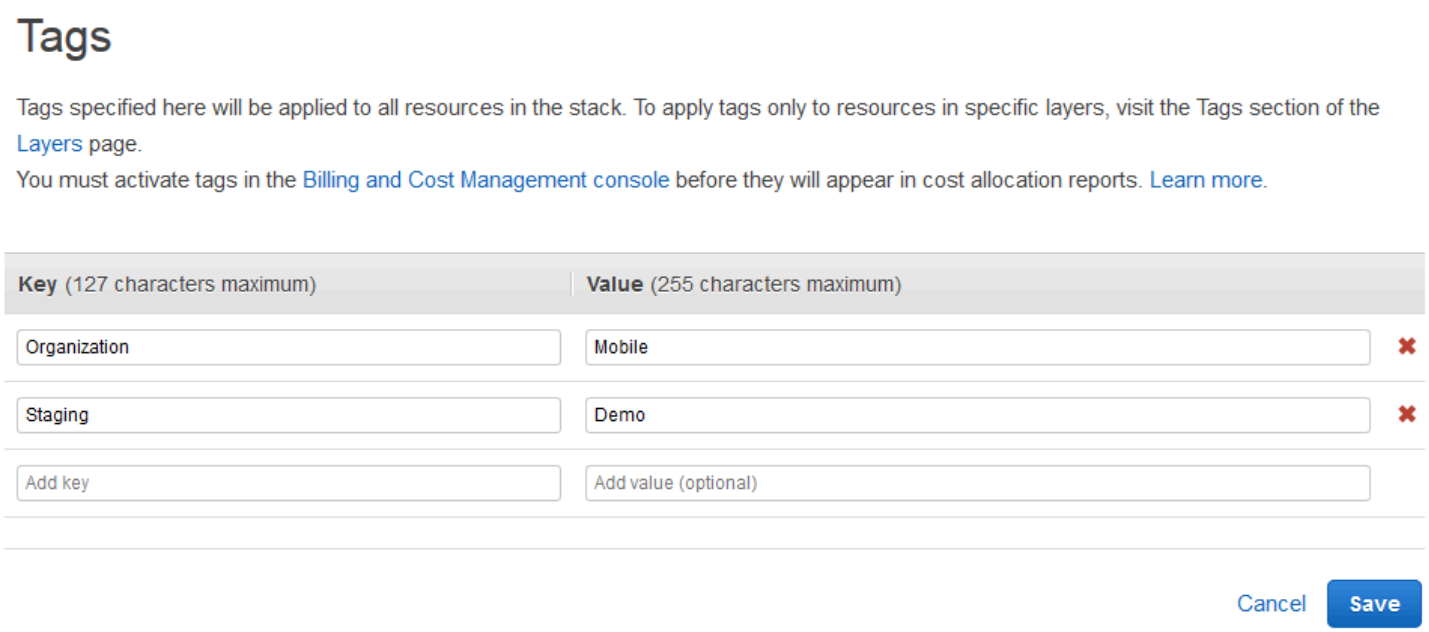
- Configuration des balises au niveau de la pile
- Configuration des balises au niveau d'une couche
- Gérer les tags à l'aide du AWS CLI
- Limitations des balises

Configuration des balises au niveau de la pile

Au niveau de la pile, vous pouvez ajouter et gérer des balises en choisissant Balises sur la page d'accueil de la pile.

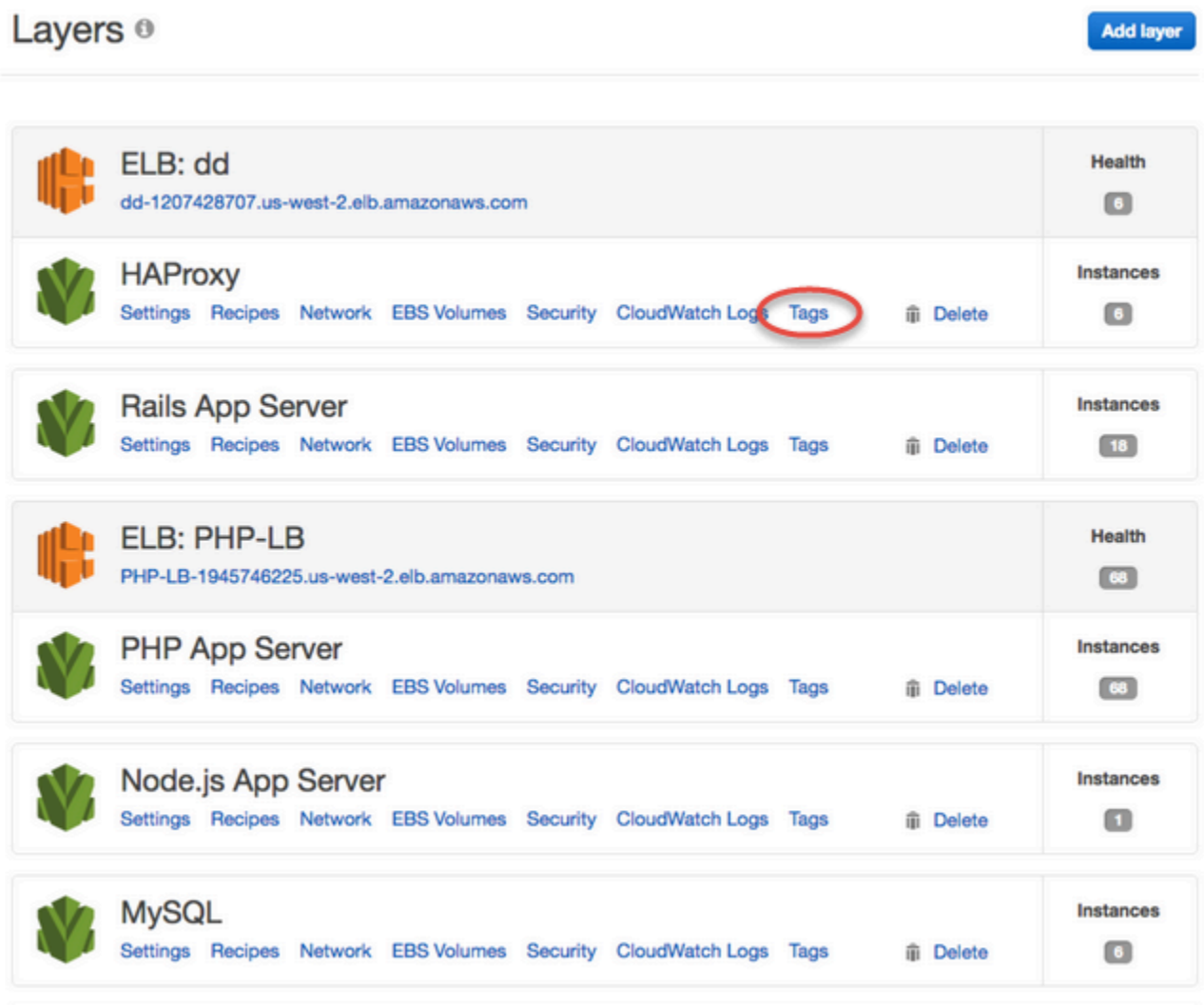


Sur la page Balises, ajoutez des balises sous forme de paires clé-valeur. La capture d'écran suivante montre des exemples de balises. Vous pouvez supprimer des balises en sélectionnant le X rouge à droite d'une paire clé-valeur.

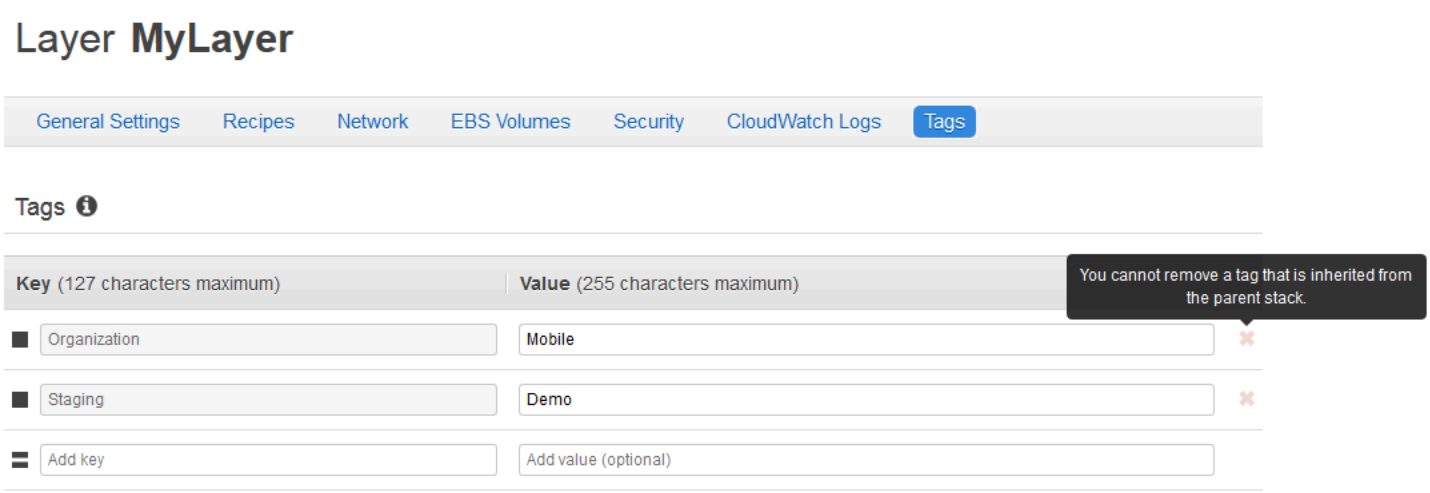


Configuration des balises au niveau d'une couche

Au niveau d'une couche, définissez des balises en choisissant l'onglet Balises. Vous pouvez trouver cet onglet sur la page d'accueil Layers (Couches) et la page d'accueil de chaque couche.



Lorsque vous modifiez ou ajoutez des balises au niveau d'une couche, gardez à l'esprit que les balises qui ont été ajoutées au niveau de la pile parent sont héritées par la couche et ses ressources. Si vous pouvez modifier les valeurs de balises héritées, vous ne pouvez pas modifier les noms de clé, ou supprimer des balises héritées. Modifiez les noms de clé ou supprimez les balises héritées de la pile parent dans les paramètres de la pile. La capture d'écran suivante montre des exemples de balises héritées du niveau de la pile. Les balises héritées sont grisées.



Pour plus d'informations sur l'ajout de balises à des piles, consultez Créer une pile. Pour plus d'informations sur l'ajout de balises à des couches, consultez Modification de OpsWorks la configuration d'une couche.

Gérer les tags à l'aide du AWS CLI

Vous pouvez également utiliser des AWS CLI commandes pour ajouter et supprimer des balises au niveau de la pile et de la couche. Pour plus d'informations sur le téléchargement et l'installation de AWS CLI, voir Installation de l'interface de ligne de AWS commande. N'oubliez pas d'ajouter le paramètre `--region` à votre commande si la pile que vous voulez baliser n'est pas dans votre région par défaut. La couche ARNs n'apparaît pas actuellement dans le AWS Management Console. Pour obtenir l'ARN d'une couche, exécutez la commande describe-layers.

Pour ajouter des balises à l'aide du AWS CLI

- À l'invite de AWS CLI commande, tapez la commande suivante, en remplaçant ***stack_or_layer_ARN*** et en spécifiant vos balises de paire clé-valeur, puis appuyez sur Entrée. L'échappement des guillemets est effectué avec des barres obliques inverses.

  ```
  aws opsworks tag-resource --resource-arn stack_or_layer_ARN --tags "{\"key\":
  \"value\",\"key\":\"value\"}"
  ```

  Voici un exemple.

  ```
  aws opsworks tag-resource --resource-arn arn:aws:opsworks:us-
  east-2:800000000003:stack/500b99c0-ec00-4cgg-8a0d-1000000jjd1b --tags "{\"Stage\":
  \"Production\",\"Organization\":\"Mobile\"}"
  ```

Pour supprimer des balises à l'aide du AWS CLI

- À l'invite de AWS CLI commande, tapez ce qui suit, puis appuyez sur Entrée.

```
aws opsworks untag-resource --resource-arn stack_or_layer_ARN --tag-keys "[\"key\",
\"key\"]"
```

Pour supprimer des balises, vous spécifiez uniquement la clé de la balise que vous voulez supprimer. Voici un exemple.

```
aws opsworks untag-resource --resource-arn arn:aws:opsworks:us-
east-2:800000000003:stack/500b99c0-ec00-4cgg-8a0d-1000000jjd1b --tag-keys "[\"Stage
\",\"Organization\"]"
```

> **Note**
>
> Vous ne pouvez pas supprimer des balises héritées (des balises qui ont été ajoutées au niveau de la pile parent) d'une couche. Vous devez supprimer les balises héritées à partir de la pile.

Limitations des balises

Gardez les limitations suivantes à l'esprit lorsque vous créez des balises.

- AWS OpsWorks Stacks limite le nombre de balises définies par l'utilisateur au niveau de la pile et de la couche à 40, y compris les balises définies par l'utilisateur héritées d'un niveau parent. Cela laisse 10 emplacements disponibles pour les balises par défaut qui sont précédées et `opsworks:` les balises définies par d'autres AWS processus. Un maximum de 50 balises est autorisé sur une ressource, y compris les balises définies par l'utilisateur et les balises par défaut créées par AWS.
- Les clés de balise ne peuvent pas commencer par **`aws:`**, **`opsworks:`** ou **`rds:`**. Ne pas utiliser **`name`** ou **`Name`** comme clé de tag, car elle **`Name`** est réservée par AWS OpsWorks Stacks.
- Une clé peut comporter 127 caractères au maximum et ne peut contenir que des lettres, des chiffres ou séparateurs Unicode, ou les caractères spéciaux suivants : `+ - = . _ : / .`
- Une valeur peut comporter 255 caractères au maximum et ne peut contenir que des lettres, des chiffres ou séparateurs Unicode, ou les caractères spéciaux suivants : `+ - = . _ : / .`

Surveillance

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez surveiller vos piles de différentes façons.

- AWS OpsWorks Stacks utilise Amazon CloudWatch pour fournir treize métriques personnalisées avec une surveillance détaillée pour chaque instance de la pile.
- AWS OpsWorks Stacks s'intègre AWS CloudTrail à Stacks pour enregistrer chaque appel d'API AWS OpsWorks Stacks et stocker les données dans un compartiment Amazon S3.
- Vous pouvez utiliser Amazon CloudWatch Logs pour surveiller le système, l'application et les journaux personnalisés de votre stack.

Rubriques

- Surveillance de Stacks à l'aide d'Amazon CloudWatch
- Logging AWS OpsWorks Stacks API avec AWS CloudTrail
- Utilisation d'Amazon CloudWatch Logs avec AWS OpsWorks Stacks
- Surveillance des piles à l'aide d'Amazon Events CloudWatch

Surveillance de Stacks à l'aide d'Amazon CloudWatch

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks utilise Amazon CloudWatch (CloudWatch) pour surveiller les piles.

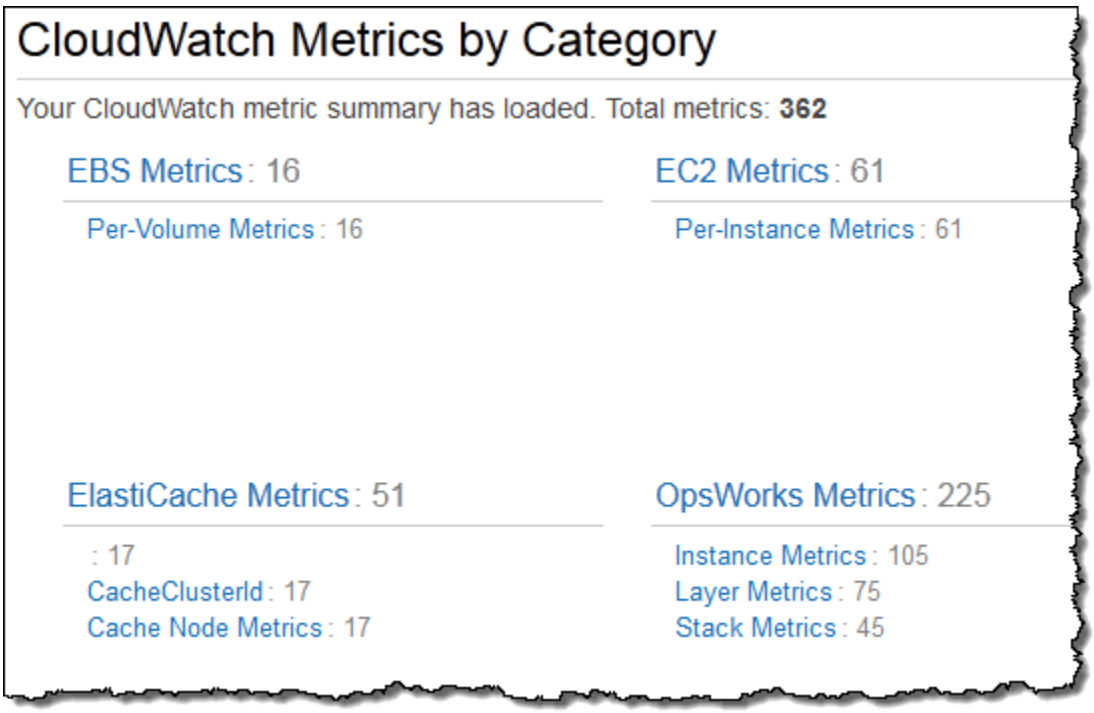
- Pour les piles Linux, AWS OpsWorks Stacks prend en charge treize métriques personnalisées pour fournir une surveillance détaillée de chaque instance de la pile et résume les données pour vous faciliter la tâche sur la page de surveillance.
- Pour Windows Stacks, vous pouvez surveiller les EC2 métriques Amazon standard pour vos instances à l'aide de la CloudWatch console.

  La page Surveillance n'affiche pas les métriques Windows.

La page Monitoring affiche les métriques d'une pile complète, d'une couche ou d'une instance. AWS OpsWorks Les métriques Stacks sont distinctes des EC2 métriques Amazon. Vous pouvez également activer des métriques supplémentaires via la CloudWatch console, mais elles nécessitent généralement des frais supplémentaires. Vous pouvez également consulter les données sous-jacentes sur la CloudWatch console, comme suit :

Pour afficher les statistiques OpsWorks personnalisées dans CloudWatch

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Sur la barre de navigation, sélectionnez la région de la pile.
3. Dans le panneau de navigation, sélectionnez Métriques.
4. Dans OpsWorks Metrics, sélectionnez Instance Metrics, Layer Metrics ou Stack Metrics.



> **Note**
>
> AWS OpsWorks Stacks collecte des métriques en exécutant un processus sur chaque instance (l'agent d'instance). Comme les métriques sont CloudWatch collectées différemment, à l'aide de l'hyperviseur, les valeurs de la CloudWatch console peuvent légèrement différer des valeurs correspondantes sur la page de surveillance de la console AWS OpsWorks Stacks.

Vous pouvez également utiliser CloudWatch la console pour définir des alarmes. Pour plus d'informations sur la création d'alarmes, consultez Creating Amazon CloudWatch Alarms. Pour obtenir la liste des métriques CloudWatch personnalisées, consultez AWS OpsWorks Metrics and Dimensions. Pour plus d'informations, consultez Amazon CloudWatch.

Rubriques

- AWS OpsWorks Métriques de Stacks
- Dimensions pour AWS OpsWorks Stacks Metrics
- Métriques de pile
- Métriques de couche
- Métriques des instances

AWS OpsWorks Métriques de Stacks

AWS OpsWorks Stacks envoie les métriques suivantes CloudWatch toutes les cinq minutes.

Métriques du processeur

Métrique	Description
`cpu_idle`	Pourcentage de temps durant lequel l'UC est inactive. Dimensions valides : IDs les ressources individuelles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune
`cpu_nice`	Pourcentage de temps pendant lequel le processeur gère les processus ayant une `nice` valeur positive, dont la priorité de planification est inférieure. Pour plus d'informations sur ce que cela permet de mesurer, consultez nice (Unix). Dimensions valides : IDs les ressources individuelles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune
`cpu_steal`	À mesure qu'AWS alloue les ressources du processeur de l'hyperviseur à un nombre croissant d'instances, la charge de virtualisation augmente et

Métrique	Description
	peut affecter la fréquence à laquelle l'hyperviseur peut effectuer le travail demandé sur une instance. `cpu_steal` mesure le pourcentage de temps pendant lequel une instance attend que l'hyperviseur alloue des ressources CPU physiques. Dimensions valides : IDs les ressources individuelles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune
`cpu_system`	Pourcentage de temps pendant lequel le processeur gère les opérations du système. Dimensions valides : IDs les ressources individuelles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune

Métrique	Description
`cpu_user`	Pourcentage de temps pendant lequel le processeur gère les opérations utilisateur. Dimensions valides : IDs les ressources individuelles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune
`cpu_waitio`	Pourcentage de temps pendant lequel le processeur attend les opérations d'entrée/sortie. Dimensions valides : IDs les ressources individuelles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune

Métriques de mémoire

Métrique	Description
`memory_buffers`	La quantité de mémoire tampon. Dimensions valides : IDs les ressources individuelles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId.

Métrique	Description
	Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune
`memory_cached`	La quantité de mémoire mise en cache. Dimensions valides : IDs les ressources individuelles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune
`memory_free`	La quantité de mémoire libre. Dimensions valides : IDs les ressources individuelles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune

Métrique	Description
`memory_swap`	La quantité d'espace de swap. Dimensions valides : IDs les ressources individuelles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune
`memory_total`	Quantité totale de mémoire. Dimensions valides : IDs les ressources individuelles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune
`memory_used`	La quantité de mémoire utilisée. Dimensions valides : IDs les ressources individuelles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune

Métriques de charge

Métrique	Description
`load_1`	La charge a été calculée en moyenne sur une période d'une minute. Dimensions valides : IDs les ressources individue lles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune
`load_5`	La charge a été calculée en moyenne sur une période de cinq minutes. Dimensions valides : IDs les ressources individue lles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune
`load_15`	La charge a été calculée en moyenne sur une fenêtre de 15 minutes. Dimensions valides : IDs les ressources individue lles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`.

Métrique	Description
	Unité : aucune

Métriques du processus

Métrique	Description
`procs`	Le nombre de processus actifs. Dimensions valides : IDs les ressources individue lles pour lesquelles vous consultez les métriques : StackId LayerId, ou InstanceId. Statistiques valides : `Average`,`Minimum`,`Maximum`,`Sum`, ou`Data Samples`. Unité : aucune

Dimensions pour AWS OpsWorks Stacks Metrics

AWS OpsWorks Les métriques Stacks utilisent l'espace de noms AWS OpsWorks Stacks et fournissent des métriques pour les dimensions suivantes :

Dimension	Description
`StackId`	Valeurs moyennes pour une pile.
`LayerId`	Valeurs moyennes pour une couche.
`InstanceId`	Valeurs moyennes pour une instance.

Métriques de pile

Pour afficher un résumé des métriques pour une pile complète, sélectionnez une pile dans le tableau de bord AWS OpsWorks Stacks, puis cliquez sur Monitoring dans le volet de navigation. L'exemple suivant concerne une pile avec une couche PHP et une couche DB.



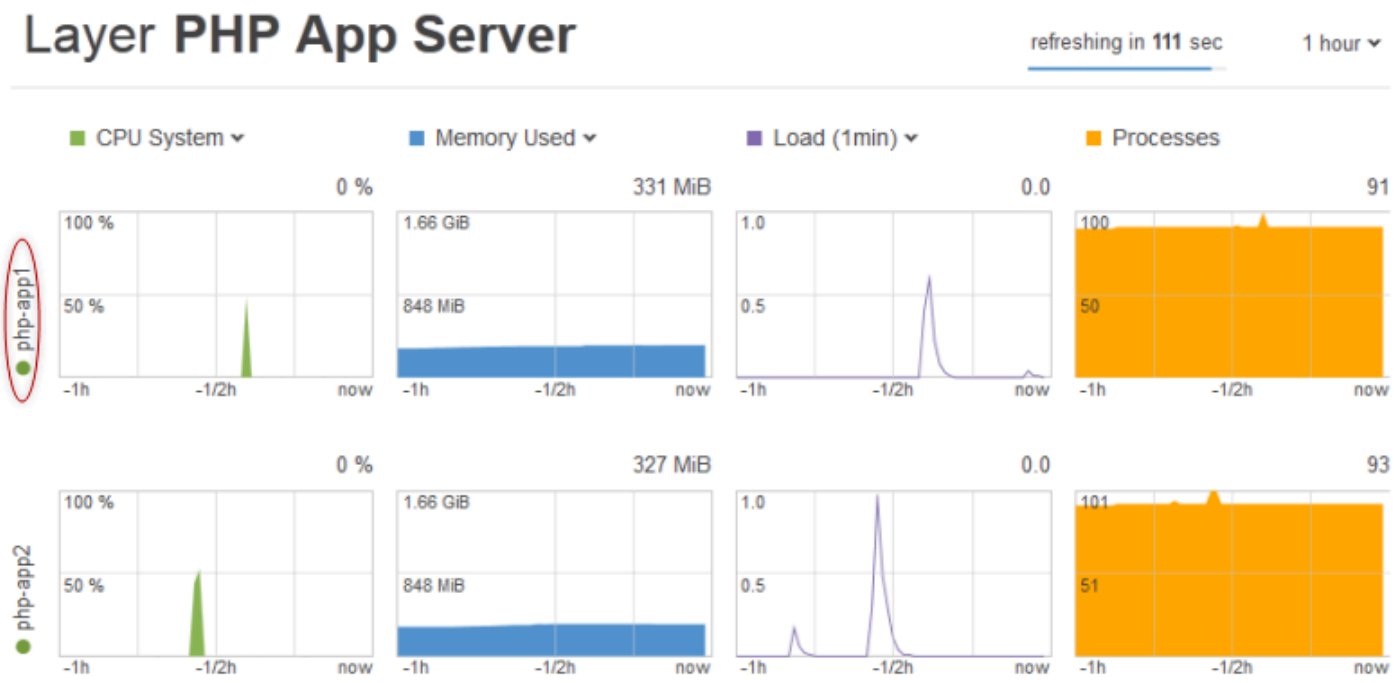
La vue de pile affiche les graphiques des quatre types de métriques de chaque couche sur une durée déterminée : 1 heure, 8 heures, 24 heures, 1 semaine ou 2 semaines. Remarques :

- AWS OpsWorks Stacks met régulièrement à jour les graphiques ; le compte à rebours en haut à droite indique le temps restant avant la prochaine mise à jour,
- Si une couche comporte plusieurs instances, les graphiques affichent les valeurs moyennes de la couche.
- Vous pouvez spécifier la période en cliquant sur la liste dans le coin supérieur droit et en sélectionnant votre valeur préférée.

Pour chaque type de métrique, vous pouvez utiliser la liste en haut du graphique et sélectionner la métrique particulière que vous souhaitez afficher.

Métriques de couche

Pour afficher les métriques d'une couche particulière, cliquez sur le nom de la couche dans la vue Monitoring Layers (Surveillance des couches). L'exemple suivant affiche les métriques de la couche PHP, qui possède deux instances.

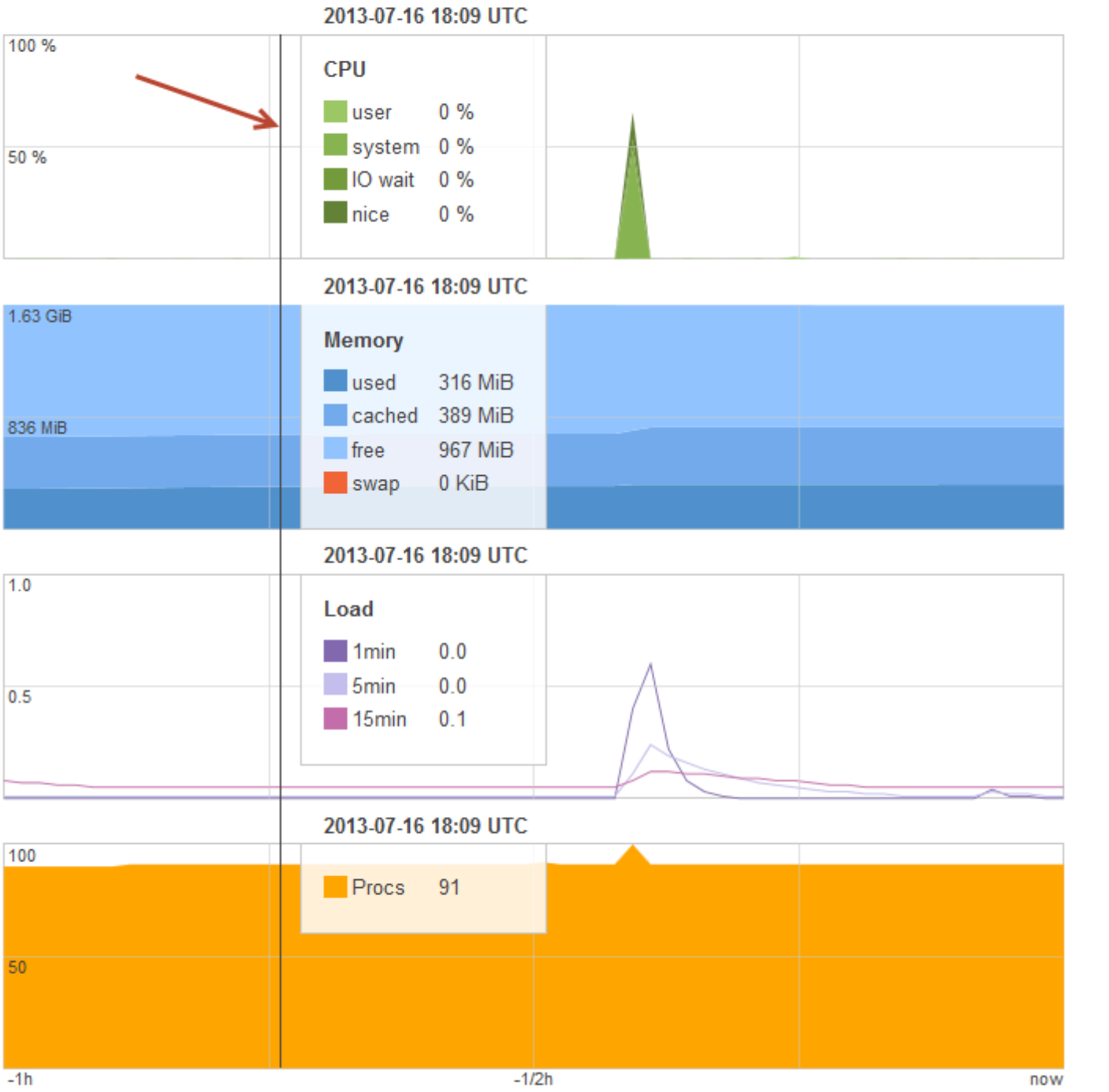


Les types de métriques sont les mêmes que pour les métriques de pile ; et pour chaque type, vous pouvez utiliser la liste en haut du graphique pour sélectionner la métrique particulière que vous souhaitez afficher.

> **Note**
>
> Vous pouvez également afficher les métriques de couche en allant à la page des détails de la couche et en cliquant sur Surveillance dans le coin supérieur droit.

Métriques des instances

Pour afficher les métriques d'une instance particulière, cliquez sur le nom de l'instance dans la vue de supervision de la couche. L'exemple suivant affiche les métriques de l'instance php-app1 de la couche PHP.


Instance php-app1
refreshing in
2013-07-16 18:09 UTC
100 %
50 %
CPU
user 0 %
system 0 %
IO wait 0 %
nice 0 %
2013-07-16 18:09 UTC
1.63 GiB
836 MiB
Memory
used 316 MiB
cached 389 MiB
free 967 MiB
swap 0 KiB
2013-07-16 18:09 UTC
1.0
0.5
Load
1min 0.0
5min 0.0
15min 0.1
2013-07-16 18:09 UTC
100
50
Procs 91
-1h
-1/2h
now

Les graphiques récapitulent toutes les métriques disponibles pour chaque type de métrique. Pour obtenir les valeurs exactes d'un point dans le temps particulier, utilisez votre souris pour déplacer le curseur (indiqué par la flèche rouge dans l'illustration précédente) jusqu'à la position appropriée.

> **Note**
>
> Vous pouvez également afficher les métriques d'instance en allant à la page des détails de l'instance et en choisissant Surveillance dans le coin supérieur droit.

Logging AWS OpsWorks Stacks API avec AWS CloudTrail

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par une identité IAM ou à un AWS service dans AWS OpsWorks Stacks. CloudTrail capture tous les appels d'API pour AWS OpsWorks Stacks sous forme d'événements, y compris les appels depuis la console AWS OpsWorks Stacks et les appels de code vers les AWS OpsWorks Stacks. APIs Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour AWS OpsWorks Stacks. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite à AWS OpsWorks Stacks, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des détails supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

AWS OpsWorks Empile les informations dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité se produit dans AWS OpsWorks Stacks, cette activité est enregistrée dans un CloudTrail événement avec

d'autres événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre AWS compte. Pour plus d'informations, consultez la section Affichage des événements avec l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements pour AWS OpsWorks Stacks, créez un parcours. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez :

- Présentation de la création d'un journal d'activité
- CloudTrail Services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

Toutes les actions de AWS OpsWorks Stacks sont enregistrées CloudTrail et documentées dans la référence de l'API AWS OpsWorks Stacks. Par exemple, les appels aux `CreateLayerDescribeInstances`, et `StartInstance` les actions génèrent des entrées dans les fichiers CloudTrail journaux.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec des informations d'identification d'utilisateur root ou IAM.
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour plus d'informations, consultez la section Élément userIdentity CloudTrail .

Comprendre les AWS OpsWorks entrées du fichier journal de Stacks

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`CreateLayer`action.


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

Utilisation d'Amazon CloudWatch Logs avec AWS OpsWorks Stacks

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Pour simplifier le processus de surveillance des journaux sur plusieurs instances, AWS OpsWorks Stacks prend en charge Amazon CloudWatch Logs. Vous activez CloudWatch les journaux au niveau de la couche dans AWS OpsWorks Stacks. CloudWatch L'intégration des journaux fonctionne avec les stacks basés sur Linux Chef 11.10 et Chef 12. L'activation des CloudWatch journaux entraîne des frais supplémentaires. Consultez donc CloudWatchles tarifs Amazon avant de commencer.

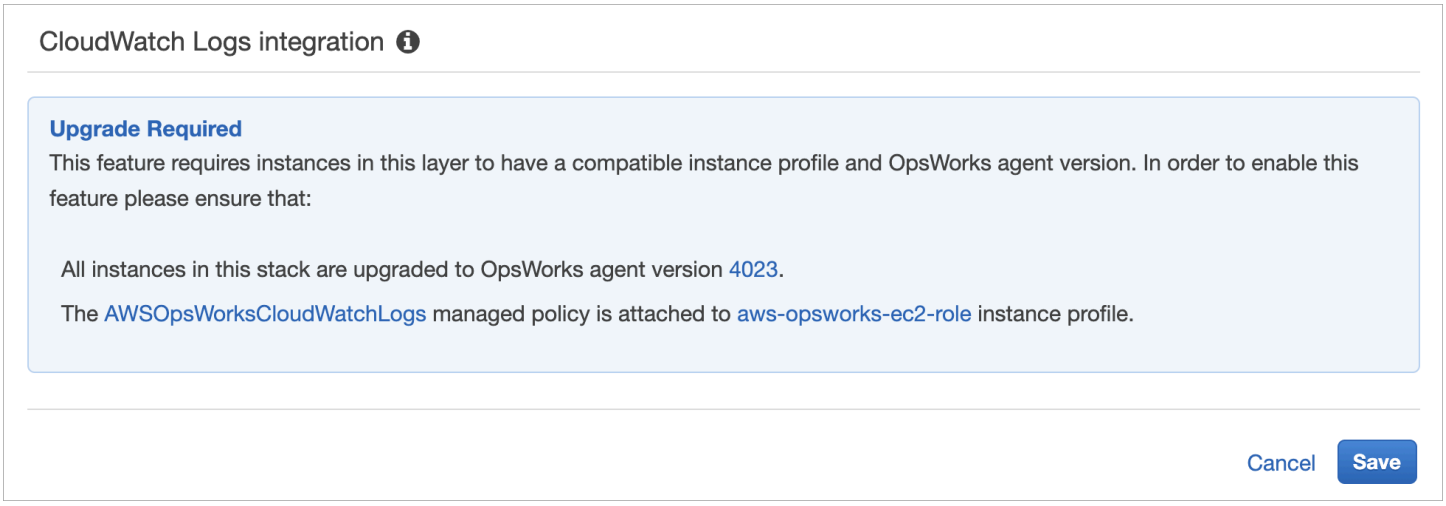
CloudWatch Logs surveille les journaux sélectionnés pour détecter l'apparition d'un modèle spécifié par l'utilisateur. Par exemple, vous pouvez surveiller les journaux afin de détecter la présence d'un terme littéral comme `NullReferenceException` ou de compter le nombre de tels événements. Une fois que vous avez activé CloudWatch Logs in AWS OpsWorks Stacks, l'agent AWS OpsWorks Stacks envoie les journaux à CloudWatch Logs. Pour plus d'informations sur CloudWatch les journaux, consultez Getting Started with CloudWatch Logs.

Prérequis

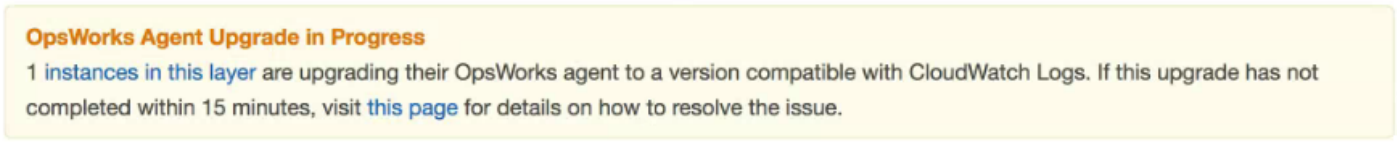
Avant de pouvoir activer CloudWatch Logs, vos instances doivent exécuter la version 3444 ou ultérieure de l'agent AWS OpsWorks Stacks dans les piles Chef 11.10, et la version 4023 ou ultérieure dans les piles Chef 12. Vous devez également utiliser un profil d'instance compatible pour toutes les instances que vous surveillez à l'aide CloudWatch des journaux.

Si vous utilisez un profil d'instance personnalisé (un profil que AWS OpsWorks Stacks n'a pas fourni lorsque vous avez créé la pile), AWS OpsWorks Stacks ne peut pas automatiquement mettre à niveau le profil d'instance. Vous devez associer manuellement la AWSOpsWorksCloudWatchLogspolitique à votre profil à l'aide d'IAM. Pour plus d'informations, consultez la section Gestion des politiques IAM dans le Guide de l'utilisateur IAM.

Si vous devez mettre à niveau la version de votre agent ou le profil de votre instance, AWS OpsWorks Stacks affiche un rappel similaire à la capture d'écran suivante lorsque vous ouvrez l'onglet CloudWatch Logs sur la page Layer.



La mise à jour de l'agent sur toutes les instances d'une couche peut prendre un peu de temps. Si vous essayez d'activer CloudWatch les journaux sur une couche avant que la mise à niveau de l'agent ne soit terminée, un message similaire au suivant s'affiche.



Activation CloudWatch des journaux

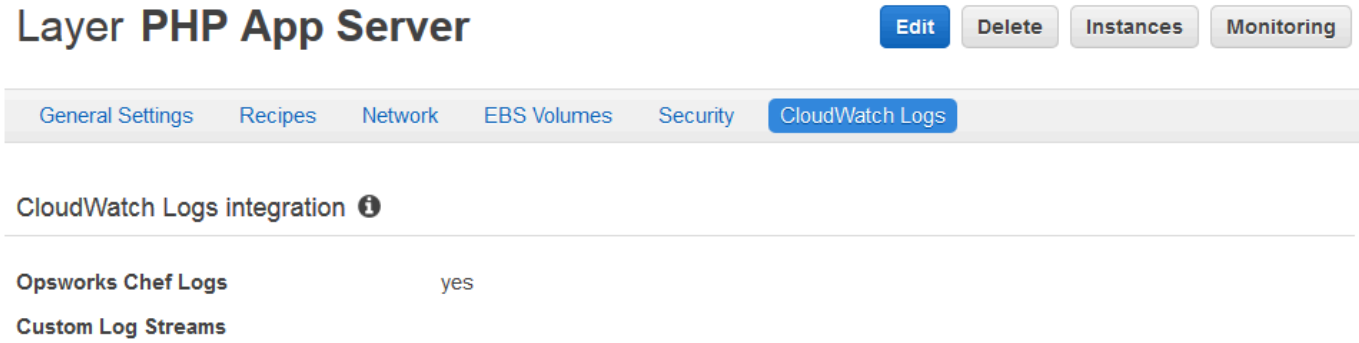
1. Une fois que les mises à niveau requises des profils d'agent et d'instance sont terminées, vous pouvez activer les CloudWatch journaux en réglant le curseur de l'onglet CloudWatch Logs sur Activé.



2. Pour diffuser les journaux de commande, définissez le curseur Diffuser les journaux de commande sur Activé. Cela envoie à Logs les journaux des activités de Chef et des commandes initiées par l'utilisateur sur les instances de CloudWatch votre couche.

Les données incluses dans ces journaux correspondent étroitement à ce que vous voyez dans les résultats d'une DescribeCommandsopération lorsque vous ouvrez la cible de l'URL du journal. Cela comprend les données sur `setup`, `configure`, `deploy`, `undeploy`, `start`, `stop`, et les commandes d'exécution de recette.

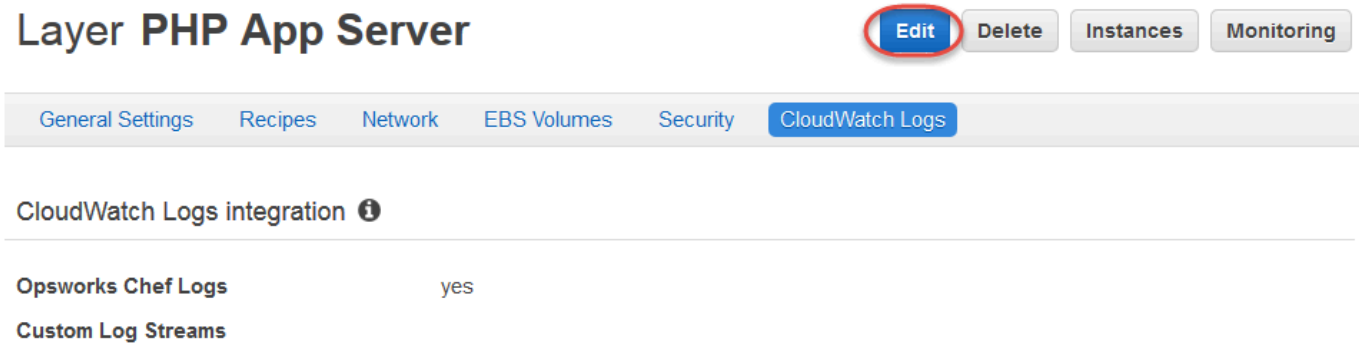
3. Pour diffuser les journaux d'activités qui sont stockés sur les instances de votre couche, comme `/var/log/apache/myapp/mylog*`, tapez l'emplacement personnalisé dans la zone Stream custom logs, puis choisissez Add (+).
4. Choisissez Save (Enregistrer). Au bout de quelques minutes, les flux de log de AWS OpsWorks Stacks devraient être visibles dans la console CloudWatch Logs.



Désactiver les CloudWatch journaux

Pour désactiver CloudWatch les journaux, modifiez les paramètres de votre couche.

1. Sur la page des propriétés de votre couche, choisissez Modifier.



2. Sur la page d'édition, choisissez l'onglet CloudWatch Logs.
3. Dans la zone CloudWatch Journaux, désactivez les journaux des commandes Stream. Sélectionnez X sur des journaux personnalisés afin de les supprimer des flux de journaux, le cas échéant.

4. Choisissez Save (Enregistrer).

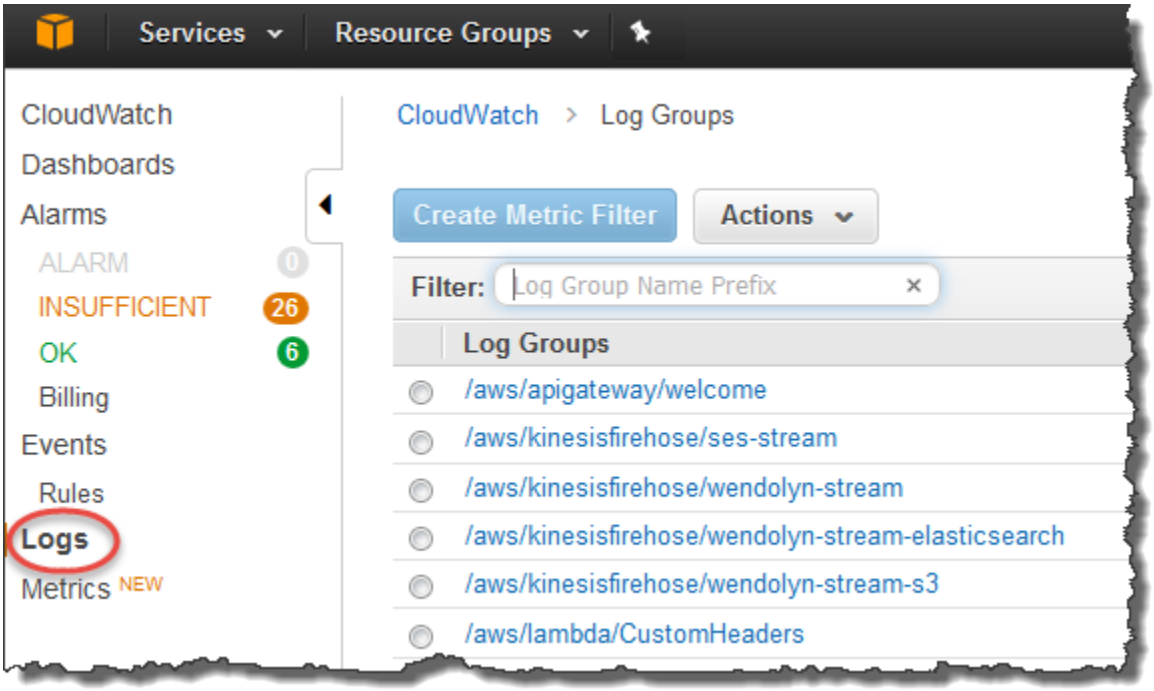
Supprimer les journaux diffusés en continu des CloudWatch journaux

Une fois que vous avez désactivé le streaming CloudWatch des journaux depuis AWS OpsWorks Stacks, les journaux existants sont toujours disponibles dans la console de gestion des CloudWatch journaux. Vous devez toujours payer des frais pour les journaux stockés, sauf si vous les exportez vers Amazon S3 ou si vous les supprimez. Pour plus d'informations sur l'exportation de journaux vers S3, consultez Exportation de données de journaux vers Amazon S3.

Vous pouvez supprimer des flux de journaux et des groupes de CloudWatch journaux dans la console de gestion des journaux ou en exécutant les `delete-log-group` AWS CLI commandes `delete-log-stream`et. Pour plus d'informations sur la modification des périodes de conservation des journaux, voir Conservation des données des journaux des modifications dans CloudWatch les journaux.

Gestion de vos connexions dans CloudWatch Logs

Les journaux que vous diffusez sont gérés dans la console CloudWatch Logs.



AWS OpsWorks crée automatiquement des groupes de journaux par défaut et des flux de journaux. Les groupes de journaux pour AWS OpsWorks Stacks ont des noms qui correspondent au modèle suivant :

stack_name/*layer_name*/*chef_log_name*

Les journaux personnalisés ont des noms qui correspondent au modèle suivant :

/stack_name/layer_short_name/file_path_name. Le nom du chemin est rendu plus lisible par l'utilisateur grâce à la suppression des caractères spéciaux, tels que les astérisques (*).

Lorsque vous avez localisé vos CloudWatch journaux dans Logs, vous pouvez les organiser en groupes, rechercher et filtrer les journaux en créant des filtres métriques et créer des alarmes personnalisées.

Configuration des couches Windows Chef 12.2 pour utiliser les journaux CloudWatch

CloudWatch L'intégration automatique des journaux n'est pas prise en charge pour les instances Windows. L'onglet CloudWatch Logs n'est pas disponible sur les couches dans les piles Chef 12.2. Pour activer manuellement le streaming vers CloudWatch les instances basées sur Logs for Windows, procédez comme suit.

- Mettez à jour le profil d'instance pour les instances Windows afin que l'agent CloudWatch Logs dispose des autorisations appropriées. La déclaration de AWSOpsWorksCloudWatchLogspolitique indique les autorisations requises.

  En général, vous n'effectuez cette tâche qu'une seule fois. Vous pouvez ensuite utiliser le profil d'instance mis à jour pour toutes les instances Windows dans une couche.
- Modifiez le fichier de configuration JSON suivant sur chaque instance. Ce fichier comprend les préférences de flux de journaux, comme les journaux à surveiller.

  ```
  %PROGRAMFILES%\Amazon\Ec2ConfigService\Settings
  \AWS.EC2.Windows.CloudWatch.json
  ```

Vous pouvez automatiser les deux tâches précédentes en créant des recettes personnalisées pour gérer les tâches requises et les attribuer aux événements Setup de la couche Chef 12.2. Chaque fois que vous démarrez une nouvelle instance sur ces couches, AWS OpsWorks Stacks exécute automatiquement vos recettes une fois le démarrage de l'instance terminé, en CloudWatch activant Logs.

Pour désactiver CloudWatch les journaux sur les instances Windows, inversez le processus. Décochez la case Activer l'intégration des CloudWatch journaux dans la boîte de dialogue Propriétés du EC2 service, supprimez les préférences de flux de journaux du `AWS.EC2.Windows.CloudWatch.json` fichier et arrêtez d'exécuter les recettes Chef qui attribuent

automatiquement CloudWatch des autorisations de journaux aux nouvelles instances des couches Chef 12.2.

Surveillance des piles à l'aide d'Amazon Events CloudWatch

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez configurer des règles dans Amazon CloudWatch Events pour vous avertir des modifications apportées aux ressources de AWS OpsWorks Stacks et demander à CloudWatch Events de prendre des mesures en fonction du contenu de l'événement. Pour plus d'informations sur la façon de démarrer avec les CloudWatch événements et de configurer les règles, consultez Getting Started with CloudWatch Events dans le guide de l'utilisateur CloudWatch des événements.

Les types d'événements AWS OpsWorks Stacks suivants sont pris en charge dans CloudWatch Events.

Instance state change (Changement d'état de l'instance)

Indique une modification de l'état d'une instance AWS OpsWorks Stacks.

Command state change (Changement d'état de la commande)

Indique qu'un changement s'est produit dans l'état d'une commande AWS OpsWorks Stacks.

Deployment state change (Changement d'état du déploiement)

Indique qu'un changement s'est produit dans l'état d'un déploiement AWS OpsWorks Stacks.

Alerts (Alertes)

Indique qu'une erreur de service AWS OpsWorks Stacks a été signalée.

Pour plus d'informations sur les types d'événements AWS OpsWorks Stacks pris en charge par les CloudWatch événements, voir AWS OpsWorks Stacks Events dans le guide de l'utilisateur CloudWatch des événements.

Sécurité et autorisations

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Chacun de vos utilisateurs doit disposer des AWS informations d'identification appropriées pour accéder aux AWS ressources de votre compte. La méthode recommandée pour fournir des informations d'identification aux utilisateurs est d'utiliser AWS Identity and Access Management(IAM). AWS OpsWorks Stacks s'intègre à IAM pour vous permettre de contrôler les éléments suivants :

- Comment les utilisateurs individuels peuvent interagir avec AWS OpsWorks Stacks.

  Par exemple, vous pouvez autoriser certains utilisateurs à déployer des applications sur n'importe quelle pile, mais pas à modifier la pile elle-même, tout en accordant à d'autres utilisateurs un accès complet, mais uniquement à certaines piles, etc.
- Comment AWS OpsWorks Stacks peut agir en votre nom pour accéder aux ressources du stack telles que les EC2 instances Amazon et les buckets Amazon S3.

  AWS OpsWorks Stacks fournit un rôle de service qui accorde des autorisations pour ces tâches.
- Comment les applications exécutées sur des EC2 instances Amazon contrôlées par AWS OpsWorks Stacks peuvent accéder à d'autres AWS ressources, telles que les données stockées dans des compartiments Amazon S3.

  Vous pouvez attribuer un profil d'instance aux instances d'une couche afin d'autoriser les applications exécutées sur ces instances à accéder à d'autres AWS ressources.
- Comment gérer les clés SSH basées sur l'utilisateur et utiliser SSH ou RDP pour se connecter aux instances.

  Pour chaque pile, les utilisateurs administratifs peuvent attribuer à chaque utilisateur une clé SSH personnelle ou autoriser les utilisateurs à spécifier leurs propres clés. Vous pouvez également autoriser un accès SSH ou RDP et des privilèges sudo ou administrateur sur les instances de la pile pour chaque utilisateur.

Les autres aspects de la sécurité sont les suivants :

- Comment gérer la mise à jour du système d'exploitation de vos instances avec les derniers correctifs de sécurité.

  Pour de plus amples informations, veuillez consulter Gestion des mises à jour de sécurité.
- Comment configurer les groupes EC2 de sécurité Amazon pour contrôler le trafic réseau à destination et en provenance de vos instances.

  Comment spécifier des groupes de sécurité personnalisés au lieu des groupes de sécurité par défaut de AWS OpsWorks Stacks. Pour de plus amples informations, veuillez consulter Utilisation des groupes de sécurité.

Rubriques

- Gestion des AWS OpsWorks autorisations utilisateur de Stacks
- Permettre à AWS OpsWorks Stacks d'agir en votre nom
- Prévention interservices confuse des adjoints dans Stacks AWS OpsWorks
- Spécification des autorisations pour les applications exécutées sur EC2 des instances
- Gestion de l'accès SSH
- Gestion des mises à jour de sécurité Linux
- Utilisation des groupes de sécurité

Gestion des AWS OpsWorks autorisations utilisateur de Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Il est recommandé de limiter les utilisateurs de AWS OpsWorks Stacks à un ensemble spécifique d'actions ou à un ensemble de ressources de stack. Vous pouvez contrôler les autorisations AWS

OpsWorks des utilisateurs de Stacks de deux manières : en utilisant la page Permissions de AWS OpsWorks Stacks et en appliquant une politique IAM appropriée.

La page OpsWorks Permissions, ou les actions équivalentes de la CLI ou de l'API, vous permet de contrôler les autorisations des utilisateurs dans un environnement multi-utilisateurs au cas par cas en attribuant à chaque utilisateur l'un des différents niveaux d'autorisation. Chaque niveau accorde les autorisations pour un ensemble standard d'actions pour une ressource de pile particulière. Sur la page Autorisations, vous pouvez contrôler les éléments suivants :

- Les personnes autorisées à accéder à chaque pile.
- Les actions que chaque utilisateur est autorisé à effectuer sur chaque pile.

  Par exemple, vous pouvez autoriser certains utilisateurs à afficher uniquement la pile, alors que d'autres peuvent déployer des applications, ajouter des instances, et ainsi de suite.
- Les personnes autorisées à gérer chaque pile.

  Vous pouvez déléguer la gestion de chaque pile à un ou plusieurs utilisateurs spécifiés.
- Qui dispose d'un accès SSH et de privilèges sudo (Linux) ou d'un accès RDP et de privilèges d'administrateur (Windows) au niveau utilisateur sur les instances Amazon de chaque pile. EC2

  Vous pouvez accorder ou supprimer ces autorisations séparément pour chaque utilisateur à tout moment.

> **Important**
>
> Le refus de l'accès SSH/RDP n'empêche pas nécessairement un utilisateur de se connecter aux instances. Si vous spécifiez une paire de EC2 clés Amazon pour une instance, tout utilisateur possédant la clé privée correspondante peut se connecter ou utiliser la clé pour récupérer le mot de passe administrateur Windows. Pour de plus amples informations, veuillez consulter Gestion de l'accès SSH.

Vous pouvez utiliser la console, la CLI ou l'API IAM pour ajouter à vos utilisateurs des politiques qui accordent des autorisations explicites pour les différentes ressources et actions de AWS OpsWorks Stacks.

- L'utilisation d'une politique IAM pour spécifier les autorisations est plus flexible que l'utilisation des niveaux d'autorisation.

- Vous pouvez configurer des identités IAM (utilisateurs, groupes d'utilisateurs et rôles), qui accordent des autorisations aux identités IAM, telles que les utilisateurs et les groupes d'utilisateurs, ou définir des rôles pouvant être associés à des utilisateurs fédérés.
- Une politique IAM est le seul moyen d'accorder des autorisations pour certaines actions clés de AWS OpsWorks Stacks.

  Par exemple, vous devez utiliser IAM pour accorder des autorisations pour `opsworks:CreateStack` et`opsworks:CloneStack`, qui sont utilisées pour créer et cloner des piles, respectivement.

Bien qu'il ne soit pas explicitement possible d'importer des utilisateurs fédérés dans la console, un utilisateur fédéré peut implicitement créer un profil utilisateur en choisissant Mes paramètres en haut à droite de la console AWS OpsWorks Stacks, puis en choisissant Utilisateurs, également en haut à droite. Sur la page Utilisateurs, les utilisateurs fédérés, dont les comptes sont créés à l'aide de l'API ou de la CLI, ou implicitement via la console, peuvent gérer leurs comptes de la même manière que les utilisateurs non fédérés.

Les deux approches ne s'excluent pas mutuellement et il est parfois utile de les combiner ; AWS OpsWorks Stacks évalue alors les deux ensembles d'autorisations. Par exemple, supposons que vous vouliez permettre aux utilisateurs d'ajouter ou de supprimer des instances, mais pas d'ajouter ou de supprimer des couches. Aucun des niveaux d'autorisation de AWS OpsWorks Stacks n'accorde cet ensemble spécifique d'autorisations. Cependant, vous pouvez utiliser la page Autorisations pour accorder aux utilisateurs un niveau d'autorisation de gestion, qui leur permet d'effectuer la plupart des opérations de stack, puis d'appliquer une politique IAM qui refuse les autorisations d'ajouter ou de supprimer des couches. Pour plus d'informations, consultez la section Contrôle de l'accès aux AWS ressources à l'aide de politiques.

Le modèle suivant est un modèle classique de gestion des autorisations utilisateur. Dans chaque cas, le lecteur (vous-même) est censé être un utilisateur administrateur.

1. Utilisez la console IAM pour appliquer des AWSOpsWorks_FullAccess politiques à un ou plusieurs utilisateurs administratifs.
2. Créez un utilisateur pour chaque utilisateur non administratif avec une politique qui n'accorde aucune autorisation AWS OpsWorks Stacks.

   Si un utilisateur a uniquement besoin d'accéder à AWS OpsWorks Stacks, il se peut que vous n'ayez pas du tout besoin d'appliquer de politique. Vous pouvez plutôt gérer leurs autorisations sur la page AWS OpsWorks Stacks Permissions.

3. Utilisez la page Utilisateurs de AWS OpsWorks Stacks pour importer les utilisateurs non administratifs dans AWS OpsWorks Stacks.
4. Pour chaque pile, utilisez la page Autorisations de la pile pour attribuer un niveau d'autorisation à chaque utilisateur.
5. Le cas échéant, personnalisez les niveaux d'autorisation des utilisateurs en appliquant une politique IAM correctement configurée.

Pour plus de recommandations sur la gestion des utilisateurs, consultezBonnes pratiques : Gestion des autorisations.

Pour plus d'informations sur les meilleures pratiques en matière d'IAM, consultez la section Bonnes pratiques de sécurité en matière d'IAM dans le guide de l'utilisateur d'IAM.

Rubriques

- Gestion des AWS OpsWorks utilisateurs de Stacks
- Octroi d'autorisations par AWS OpsWorks pile aux utilisateurs de Stacks
- Gestion des autorisations de AWS OpsWorks Stacks en joignant une politique IAM
- Exemples de stratégies
- AWS OpsWorks Stabilise les niveaux d'autorisation

Gestion des AWS OpsWorks utilisateurs de Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Avant de pouvoir importer des utilisateurs dans AWS OpsWorks Stacks et de leur accorder des autorisations, vous devez d'abord avoir créé un utilisateur pour chaque individu. Pour créer des utilisateurs IAM, commencez par vous connecter en AWS tant qu'utilisateur disposant des autorisations définies dans la politique IAMFull d'accès. Vous utilisez ensuite la console IAM pour

créer des utilisateurs IAM pour tous ceux qui ont besoin d'accéder AWS OpsWorks à Stacks. Vous pouvez ensuite importer ces utilisateurs dans AWS OpsWorks Stacks et leur accorder des autorisations comme suit :

Utilisateurs réguliers AWS OpsWorks de Stacks

Les utilisateurs réguliers ne requièrent pas une stratégie attachée. S'ils en ont une, elle n'inclut généralement aucune autorisation AWS OpsWorks Stacks. Utilisez plutôt la page AWS OpsWorks Stacks Permissions pour attribuer l'un des niveaux d'autorisation suivants aux utilisateurs réguliers sur une stack-by-stack base régulière.

- Les autorisations Afficher permettent aux utilisateurs d'afficher la pile, mais pas d'effectuer des opérations.
- Les autorisations Déployer incluent les autorisations Afficher et permettent aussi aux utilisateurs de déployer et de mettre à jour des applications.
- Les autorisations de gestion incluent les autorisations de déploiement et permettent également aux utilisateurs d'effectuer des opérations de gestion de pile, telles que l'ajout de couches ou d'instances, d'utiliser la page Autorisations pour définir les autorisations des utilisateurs et d'activer leurs propres SSH/RDP and sudo/admin privilèges.
- Les autorisations Refuser refusent l'accès à la pile.

Si ces niveaux d'autorisation ne correspondent pas exactement à ce que vous souhaitez pour un utilisateur en particulier, vous pouvez personnaliser les autorisations de l'utilisateur en appliquant une politique IAM. Par exemple, vous pouvez utiliser la page Permissions des AWS OpsWorks piles pour attribuer le niveau de gestion des autorisations à un utilisateur, qui lui accorde les autorisations nécessaires pour effectuer toutes les opérations de gestion des piles, mais pas pour créer ou cloner des piles. Vous pouvez ensuite appliquer une politique qui restreint ces autorisations en leur refusant l'autorisation d'ajouter ou de supprimer des couches ou en augmentant ces autorisations en leur permettant de créer ou de cloner des piles. Pour de plus amples informations, veuillez consulter Gestion des autorisations de AWS OpsWorks Stacks en joignant une politique IAM.

AWS OpsWorks Utilisateurs administratifs de Stacks

Les utilisateurs administratifs sont le propriétaire du compte ou un utilisateur IAM disposant des autorisations définies par la AWSOpsWorks_FullAccess politique. En plus des autorisations accordées aux utilisateurs Gérer, cette stratégie inclut les autorisations relatives aux actions qui ne peuvent pas être attribuées via la page Autorisations, telles que celles-ci :

- Importation d'utilisateurs dans AWS OpsWorks Stacks

- Création et clonage de piles

Pour obtenir la stratégie complète, consultez Exemples de stratégies. Pour une liste détaillée des autorisations qui ne peuvent être accordées aux utilisateurs qu'en appliquant une politique IAM, consultezAWS OpsWorks Stabilise les niveaux d'autorisation.

Rubriques

- Utilisateurs et régions
- Création d'un utilisateur administratif de AWS OpsWorks Stacks
- Création d'utilisateurs IAM pour Stacks AWS OpsWorks
- Importation d'utilisateurs dans AWS OpsWorks Stacks
- Modification des AWS OpsWorks paramètres utilisateur de Stacks

Utilisateurs et régions

AWS OpsWorks Les utilisateurs de Stacks sont disponibles dans le point de terminaison régional dans lequel ils ont été créés. Vous pouvez créer des utilisateurs dans l'une des régions suivantes.

- Région US East (Ohio)
- Région USA Est (Virginie du Nord)
- Région USA Ouest (Oregon)
- Région US West (N. California)
- Région du Canada (Centre) (API uniquement ; non disponible dans le AWS Management Console
- Région Asie-Pacifique (Mumbai)
- Région Asia Pacific (Singapore)
- Région Asie-Pacifique (Sydney)
- Région Asia Pacific (Tokyo)
- Région Asia Pacific (Seoul)
- Région Europe (Frankfurt)
- Région Europe (Irlande)
- Région Europe (Londres)
- Région Europe (Paris)
- Région Amérique du Sud (São Paulo)

Lorsque vous importez des utilisateurs dans AWS OpsWorks Stacks, vous les importez vers l'un des points de terminaison régionaux ; si vous souhaitez qu'un utilisateur soit disponible dans plusieurs régions, vous devez l'importer dans cette région. Vous pouvez également importer des utilisateurs AWS OpsWorks Stacks d'une région à l'autre ; si vous importez un utilisateur dans une région qui possède déjà un utilisateur portant le même nom, l'utilisateur importé remplace l'utilisateur existant. Pour plus d'informations sur l'importation d'utilisateurs, consultez Importation d'utilisateurs.

Création d'un utilisateur administratif de AWS OpsWorks Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez créer un utilisateur administratif de AWS OpsWorks Stacks en ajoutant la `AWSOpsWorks_FullAccess` politique à un utilisateur, qui accorde à cet utilisateur des autorisations d'accès complet à AWS OpsWorks Stacks. Pour plus d'informations sur la création d'un utilisateur administratif, voir Création d'un utilisateur administratif.

> **Note**
>
> La AWSOpsWorks_FullAccess politique permet aux utilisateurs de créer et de gérer des AWS OpsWorks piles Stacks, mais les utilisateurs ne peuvent pas créer de rôle de service IAM pour la pile ; ils doivent utiliser un rôle existant. Le premier utilisateur à créer une pile doit disposer d'autorisations IAM supplémentaires, comme décrit dans Autorisations administratives. Lorsque cet utilisateur crée la première pile, AWS OpsWorks Stacks crée un rôle de service IAM avec les autorisations requises. Par la suite, n'importe quel utilisateur avec les autorisations `opsworks:CreateStack` peut utiliser ce rôle pour créer des piles supplémentaires. Pour de plus amples informations, veuillez consulter Permettre à AWS OpsWorks Stacks d'agir en votre nom.

Lorsque vous créez un utilisateur, vous pouvez ajouter des politiques supplémentaires gérées par le client pour affiner les autorisations de l'utilisateur, selon les besoins. Par exemple, il se peut que vous

souhaitiez qu'un utilisateur administratif soit capable de créer ou de supprimer des piles, mais pas d'importer de nouveaux utilisateurs. Pour de plus amples informations, veuillez consulter Gestion des autorisations de AWS OpsWorks Stacks en joignant une politique IAM.

Si vous avez plusieurs utilisateurs administratifs, au lieu de définir des autorisations séparément pour chaque utilisateur, vous pouvez ajouter la AWSOpsWorks_FullAccess politique à un groupe IAM et ajouter les utilisateurs à ce groupe.

Pour plus d'informations sur la création d'un groupe, consultez la section Création de groupes d'utilisateurs IAM. Lorsque vous créez le groupe, ajoutez la AWSOpsWorks_FullAccesspolitique. Vous pouvez également ajouter la AdministratorAccesspolitique, qui inclut les AWSOpsWorks_FullAccessautorisations.

Pour plus d'informations sur l'ajout d'autorisations à un groupe existant, voir Attacher une politique à un groupe d'utilisateurs IAM.

Création d'utilisateurs IAM pour Stacks AWS OpsWorks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Avant de pouvoir importer des utilisateurs IAM dans AWS OpsWorks Stacks, vous devez les créer. Vous pouvez le faire à l'aide de la console IAM, de la ligne de commande ou de l'API. Pour obtenir des instructions complètes, consultez la section Création d'un utilisateur IAM dans votre AWS compte.

Notez que, contrairement aux utilisateurs administratifs, vous n'avez pas besoin d'attacher une stratégie pour définir des autorisations. Vous pouvez définir les autorisations après l'importation de ces utilisateurs dans AWS OpsWorks Stacks, comme indiqué dans Gestion des autorisations utilisateur.

Pour plus d'informations sur la création d'utilisateurs et de groupes IAM, consultez Getting started with IAM.

Importation d'utilisateurs dans AWS OpsWorks Stacks

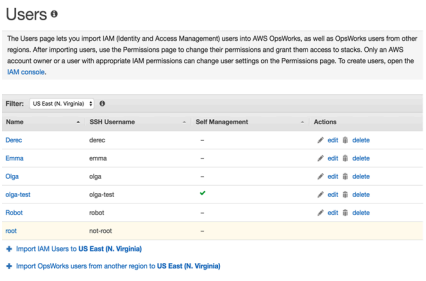
> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les utilisateurs administratifs peuvent importer des utilisateurs dans AWS OpsWorks Stacks ; ils peuvent également importer des utilisateurs AWS OpsWorks Stacks d'un point de terminaison régional à un autre. Lorsque vous importez des utilisateurs dans AWS OpsWorks Stacks, vous les importez vers l'un des points de terminaison régionaux de AWS OpsWorks Stacks. Si vous souhaitez qu'un utilisateur soit disponible dans plusieurs régions, vous devez l'importer dans cette région.

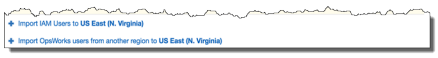
Bien qu'il ne soit pas explicitement possible d'importer des utilisateurs fédérés dans la console, un utilisateur fédéré peut implicitement créer un profil utilisateur en choisissant Mes paramètres en haut à droite de la console AWS OpsWorks Stacks, puis en choisissant Utilisateurs, également en haut à droite. Sur la page Utilisateurs, les utilisateurs fédérés, dont les comptes sont créés à l'aide de l'API ou de la CLI, ou implicitement via la console, peuvent gérer leurs comptes de la même manière que les utilisateurs non fédérés.

Pour importer des utilisateurs dans AWS OpsWorks Stacks

1. Connectez-vous à AWS OpsWorks Stacks en tant qu'utilisateur administratif ou en tant que propriétaire du compte.
2. Choisissez Utilisateurs dans le coin supérieur droit pour ouvrir la page Utilisateurs.



3. Choisissez Importer les utilisateurs IAM vers < ***region name*** > pour afficher les utilisateurs disponibles, mais qui n'ont pas encore été importés.



4. Cochez la case Sélectionner tout ou sélectionnez un ou plusieurs utilisateurs individuels. Lorsque vous avez terminé, choisissez Importer vers OpsWorks.

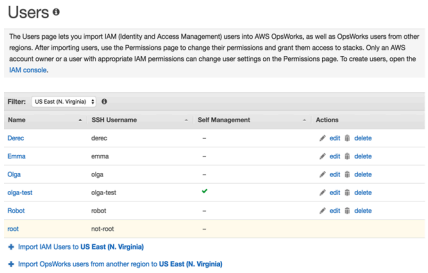
> **Note**
>
> Après avoir importé un utilisateur dans AWS OpsWorks Stacks, si vous utilisez la console ou l'API IAM pour le supprimer de votre compte, l'utilisateur ne perd pas automatiquement l'accès SSH que vous lui avez accordé via Stacks. AWS OpsWorks Vous devez également supprimer l'utilisateur de AWS OpsWorks Stacks en ouvrant la page Utilisateurs et en choisissant Supprimer dans la colonne Actions de l'utilisateur.

Pour importer des utilisateurs AWS OpsWorks Stacks d'une région à une autre

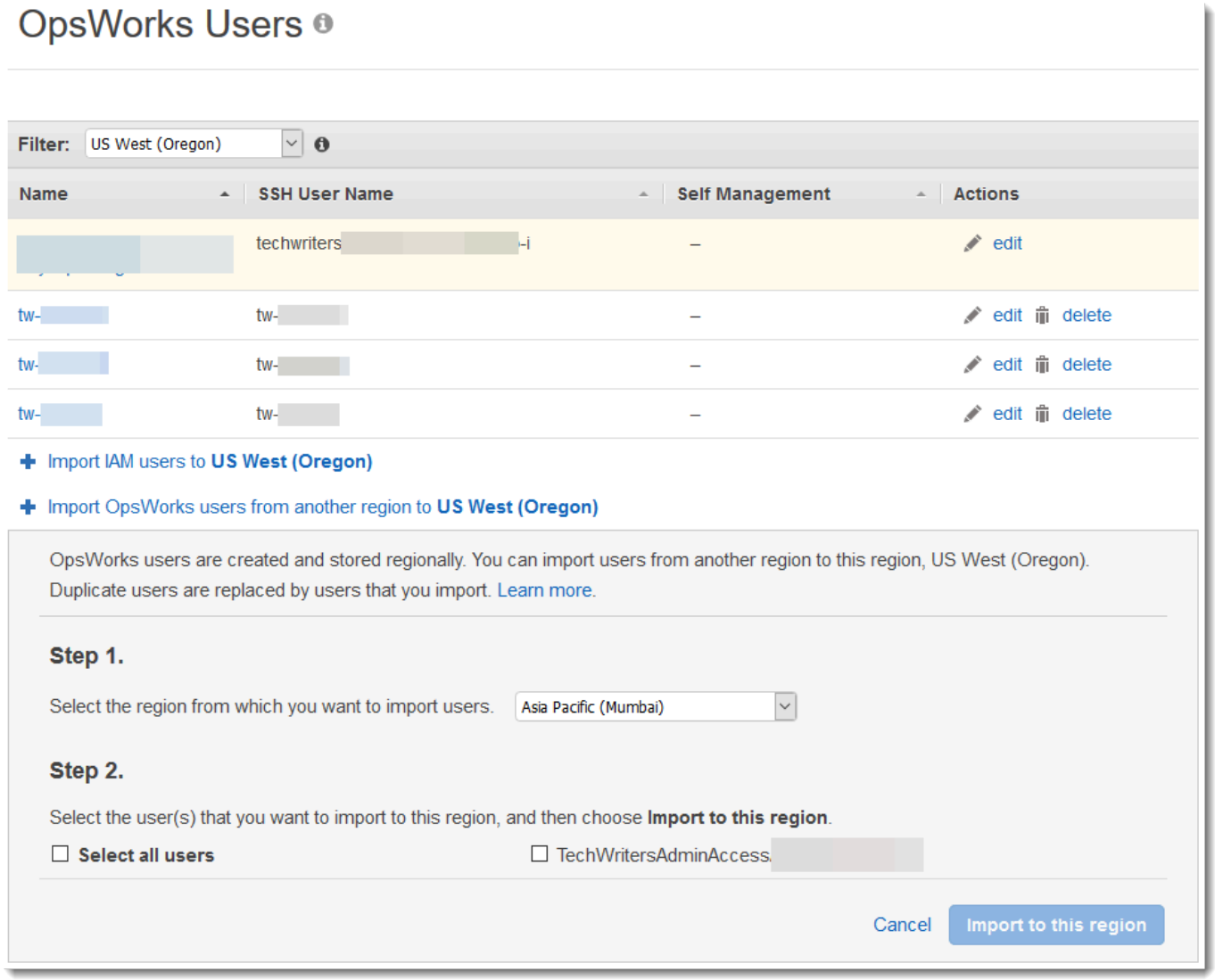
AWS OpsWorks Les utilisateurs de Stacks sont disponibles dans le point de terminaison régional dans lequel ils ont été créés. Vous pouvez créer des utilisateurs dans les régions indiquées dansUtilisateurs et régions.

Vous pouvez importer des utilisateurs AWS OpsWorks Stacks d'une région vers la région dans laquelle votre liste d'utilisateurs est actuellement filtrée. Si vous importez un utilisateur dans une région qui possède déjà un utilisateur portant le même nom, l'utilisateur importé remplace l'utilisateur existant.

1. Connectez-vous à AWS OpsWorks Stacks en tant qu'utilisateur administratif ou en tant que propriétaire du compte.
2. Choisissez Utilisateurs dans le coin supérieur droit pour ouvrir la page Utilisateurs. Si vous avez des utilisateurs de AWS OpsWorks Stacks dans plusieurs régions, utilisez le contrôle Filtrer pour filtrer en fonction de la région dans laquelle vous souhaitez importer des utilisateurs.



3. Choisissez Importer les utilisateurs de AWS OpsWorks Stacks d'une autre région vers < ***current region*** >.



4. Sélectionnez la région à partir de laquelle vous souhaitez importer des utilisateurs de AWS OpsWorks Stacks.
5. Sélectionnez un ou plusieurs utilisateurs à importer, ou sélectionnez tous les utilisateurs, puis choisissez Importer vers cette région. Attendez que AWS OpsWorks Stacks affiche les utilisateurs importés dans la liste des utilisateurs.

Unix IDs et utilisateurs créés en dehors AWS OpsWorks de Stacks

AWS OpsWorks attribue aux utilisateurs des instances AWS OpsWorks Stacks des valeurs d'identifiant Unix (UID) comprises entre 2000 et 4000. Dans la AWS OpsWorks mesure où les réserves se situent entre 2000 et 4000 UIDs, les utilisateurs que vous créez en dehors AWS OpsWorks (en utilisant des recettes de livres de recettes ou en important des utilisateurs AWS OpsWorks depuis IAM, par exemple) peuvent être remplacés UIDs par AWS OpsWorks Stacks pour

un autre utilisateur. Cela peut avoir pour conséquence que les utilisateurs que vous avez créés en dehors de AWS OpsWorks Stacks n'apparaissent pas dans les résultats de recherche dans les sacs de données ou soient exclus du fonctionnement intégré `sync_remote_users` de AWS OpsWorks Stacks.

Les processus externes peuvent également créer des utilisateurs UIDs que AWS OpsWorks Stacks peut remplacer. Certains packages de systèmes d'exploitation, par exemple, peuvent créer un utilisateur dans le cadre des processus post-installation. Lorsque vous ou un processus logiciel créez un utilisateur sur un système d'exploitation basé sur Linux sans spécifier explicitement d'UID (qui est le paramètre par défaut), l'UID attribué par Stacks est +1. AWS OpsWorks *<highest existing AWS OpsWorks UID>*

Il est recommandé de créer des utilisateurs AWS OpsWorks Stacks et de gérer leur accès dans la console AWS OpsWorks Stacks ou à l'aide d'un AWS SDK. AWS CLI Si vous créez des utilisateurs sur des instances AWS OpsWorks Stacks en dehors de AWS OpsWorks, utilisez des *UnixID* valeurs supérieures à 4 000.

Modification des AWS OpsWorks paramètres utilisateur de Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une fois que vous avez importé les utilisateurs, vous pouvez modifier leurs paramètres, comme suit :

Pour modifier les paramètres utilisateur

1. Sur la page Utilisateurs, choisissez Modifier dans la colonne Actions de l'utilisateur.
2. Vous pouvez spécifier les paramètres suivants.

   Gestion automatique

   Sélectionnez Oui pour autoriser l'utilisateur à utiliser la MySettings page pour spécifier sa clé SSH personnelle.

> **Note**
>
> Vous pouvez également activer l'autogestion en ajoutant une politique IAM à l'identité IAM qui accorde des autorisations pour les DescribeMyUserProfileactions et. UpdateMyUserProfile

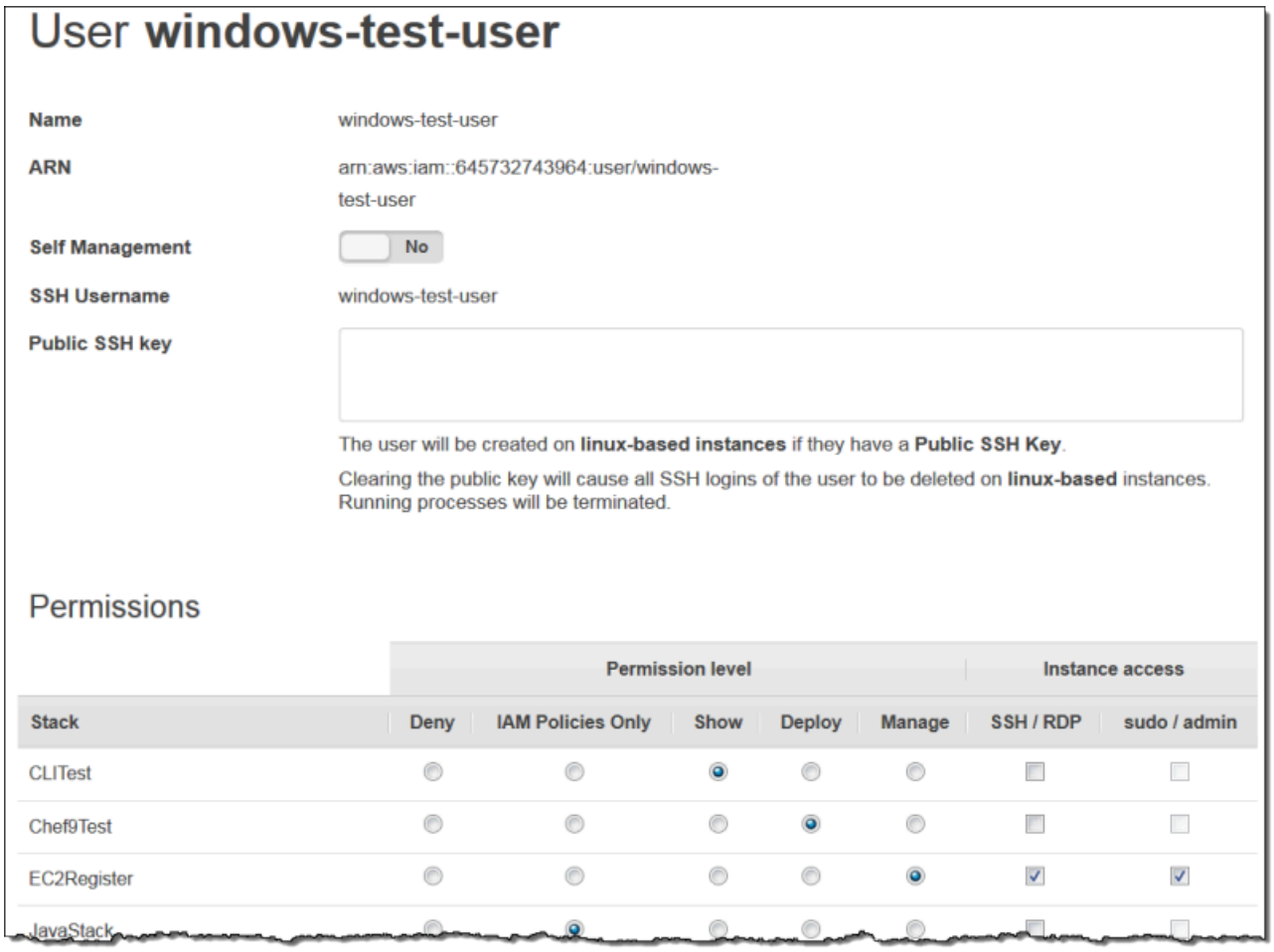
Clé SSH publique

(Facultatif) Entrez une clé publique SSH pour l'utilisateur. Cette clé s'affiche sur la page Mes paramètres de l'utilisateur. Si vous activez la gestion automatique, l'utilisateur peut modifier Mes paramètres et spécifier sa propre clé. Pour de plus amples informations, veuillez consulter Enregistrement de la clé SSH publique d'un utilisateur.

AWS OpsWorks Stacks installe cette clé sur toutes les instances Linux ; les utilisateurs peuvent utiliser la clé privée associée pour se connecter. Pour de plus amples informations, veuillez consulter Connexion avec SSH. Vous ne pouvez pas utiliser cette clé avec les piles Windows.

Autorisations

(Facultatif) Définissez les niveaux d'autorisation utilisateur pour chaque pile dans un seul emplacement au lieu de les définir séparément à l'aide de la page Autorisations de la pile. Pour plus d'informations sur les niveaux d'autorisation, consultez Attribution des autorisations par pile.



Octroi d'autorisations par AWS OpsWorks pile aux utilisateurs de Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Le moyen le plus simple de gérer les autorisations AWS OpsWorks des utilisateurs de Stacks consiste à utiliser la page d'autorisations d'une pile. Chaque pile a sa propre page, qui attribue les autorisations pour cette pile.

Vous devez être connecté tant qu'utilisateur administratif ou qu'utilisateur Gérer pour modifier les paramètres d'autorisation. La liste affiche uniquement les utilisateurs qui ont été importés dans AWS OpsWorks Stacks. Pour plus d'informations sur la façon de créer et d'importer des utilisateurs, consultez Gestion des utilisateurs.

Le niveau d'autorisation par défaut est IAM Policies Only, qui accorde aux utilisateurs uniquement les autorisations figurant dans leur politique IAM.

- Lorsque vous importez un utilisateur depuis IAM ou depuis une autre région, l'utilisateur est ajouté à la liste de toutes les piles existantes avec un niveau d'autorisation IAM Policies Only.
- Par défaut, un utilisateur que vous venez d'importer depuis une autre région n'a pas accès aux piles de la région de destination. Si vous importez des utilisateurs d'une autre région, pour leur permettre de gérer les piles dans la région de destination, des autorisations doivent leur être attribuées après avoir importé les utilisateurs.
- Lorsque vous créez une pile, tous les utilisateurs actuels sont ajoutés à la liste avec les niveaux d'autorisation Stratégies IAM uniquement.

Rubriques

- Définition des autorisations d'un utilisateur
- Affichage des autorisations
- Utilisation des clés de condition IAM pour vérifier les informations d'identification temporaires

Définition des autorisations d'un utilisateur

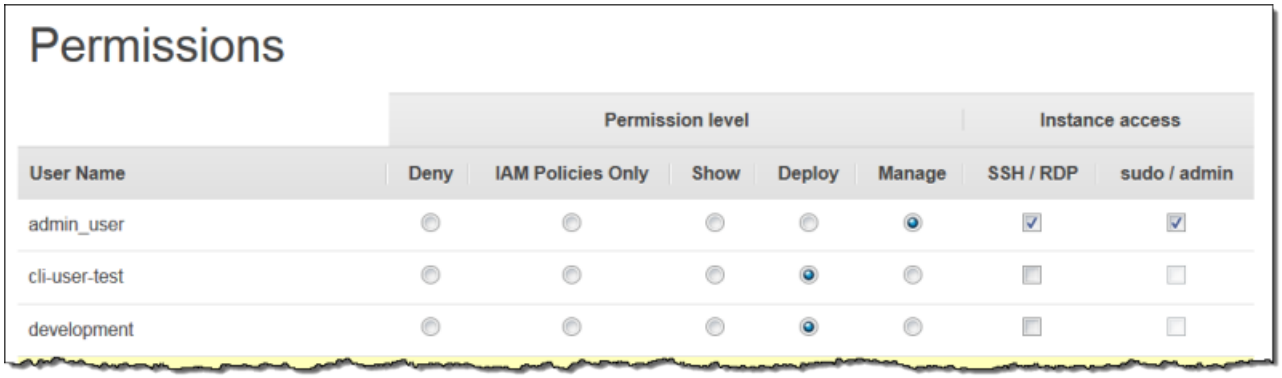
Pour définir les autorisations d'un utilisateur

1. Dans le panneau de navigation, sélectionnez Autorisations.
2. Sur la page Autorisations, choisissez Modifier.
3. Modifiez les paramètres Niveau d'autorisation et Accès à l'instance :

    - Utilisez les paramètres Niveau d'autorisation pour attribuer l'un des niveaux d'autorisation standard à chaque utilisateur, lesquels déterminent si l'utilisateur peut accéder à cette pile et les actions qu'il peut effectuer. Si un utilisateur dispose d'une politique IAM, AWS OpsWorks Stacks évalue les deux ensembles d'autorisations. Pour obtenir un exemple, consultez Exemples de stratégies.

- Le paramètre Accès à l'instance SSH/RDP spécifie si l'utilisateur dispose de l'accès SSH (Linux) ou RDP (Windows) aux instances de la pile.

  Si vous autorisez l'accès SSH/RDP, vous pouvez, le cas échéant, sélectionner sudo/admin, qui accorde à l'utilisateur les privilèges sudo (Linux) ou admin (Windows) sur les instances de la pile.



Vous pouvez attribuer à chaque utilisateur l'un des niveaux d'autorisation suivants. Pour obtenir la liste des actions autorisées par chaque niveau, consultez AWS OpsWorks Stabilise les niveaux d'autorisation.

Refuser

L'utilisateur ne peut effectuer aucune action AWS OpsWorks Stacks sur la pile, même s'il dispose d'une politique IAM qui accorde à AWS OpsWorks Stacks des autorisations d'accès complètes. Vous pouvez l'utiliser pour interdire, par exemple, à certains utilisateurs l'accès aux piles des produits non distribués.

Stratégies IAM uniquement

Niveau par défaut, assigné à tous les utilisateurs nouvellement importés et à tous les utilisateurs des piles nouvellement créées. Les autorisations de l'utilisateur sont déterminées par sa politique IAM. Si un utilisateur n'a aucune politique IAM, ou si sa politique ne comporte aucune autorisation AWS OpsWorks Stacks explicite, il ne peut pas accéder à la pile. Ce niveau est généralement attribué aux utilisateurs administratifs car leurs politiques IAM accordent déjà des autorisations d'accès complètes.

Afficher

L'utilisateur peut afficher une pile, mais ne peut pas effectuer d'opérations. Par exemple, il se peut que les gestionnaires souhaitent surveiller les piles d'un compte, mais qu'ils n'aient pas besoin de déployer des applications ou de modifier la pile de quelque façon que ce soit.

Déploiement

Inclut les autorisations Show et permet également à l'utilisateur de déployer des applications. Par exemple, un développeur d'application peut avoir besoin de déployer des mises à jour sur les instances de la pile, mais pas d'ajouter des couches ou des instances à la pile.

Gérer

Inclut les autorisations Deploy et permet également à l'utilisateur d'effectuer diverses opérations de gestion de la pile, y compris les actions suivantes :

- Ajout ou suppression de couches et d'instances.
- Utilisation de la pile Permissions pour attribuer les niveaux d'autorisation aux utilisateurs.
- Enregistrement ou annulation de l'enregistrement des ressources.

Par exemple, chaque pile peut avoir un gestionnaire dédié, chargé de s'assurer que la pile possède un nombre et un type appropriés d'instances, de gérer les mises à jour des packages et du système d'exploitation, et ainsi de suite.

> **Note**
>
> Le niveau Manage ne permet pas aux utilisateurs de créer des piles ou d'en cloner. Ces autorisations doivent être accordées par une politique IAM. Pour obtenir un exemple, consultez Gestion des autorisations.

Si l'utilisateur dispose également d'une politique IAM, AWS OpsWorks Stacks évalue les deux ensembles d'autorisations. Cela vous permet d'attribuer un niveau d'autorisation à un utilisateur, puis d'appliquer une politique pour restreindre ou augmenter les actions autorisées au niveau. Par exemple, vous pouvez appliquer une politique qui autorise un utilisateur de gestion à créer ou à cloner des piles, ou qui refuse à cet utilisateur la possibilité d'enregistrer ou de désenregistrer des ressources. Pour obtenir des exemples de ces stratégies, consultez Exemples de stratégies.

> **Note**
>
> Si la stratégie de l'utilisateur autorise des actions supplémentaires, le résultat peut apparaître pour remplacer les paramètres de la page Permissions. Par exemple, si un utilisateur dispose d'une politique qui autorise l'CreateLayeraction mais que vous utilisez la page Autorisations pour spécifier les autorisations de déploiement, l'utilisateur est toujours autorisé à créer des couches. L'exception à cette règle est l'option Deny, qui refuse l'accès à la pile même aux utilisateurs soumis à des AWSOpsWorks_FullAccess politiques. Pour plus d'informations, consultez la section Contrôle de l'accès aux AWS ressources à l'aide de politiques.

Affichage des autorisations

Si l'option self-management est activée, les utilisateurs peuvent afficher un résumé de leurs niveaux d'autorisations pour chaque pile en choisissant My Settings, dans l'angle supérieur droit. Les utilisateurs peuvent également accéder à Mes paramètres si leur politique accorde des autorisations pour les UpdateMyUserProfileactions DescribeMyUserProfileet.

Utilisation des clés de condition IAM pour vérifier les informations d'identification temporaires

AWS OpsWorks Stacks possède une couche d'autorisation intégrée qui prend en charge les cas d'autorisation supplémentaires (tels que la gestion simplifiée de l'accès en lecture seule ou en lecture-écriture aux piles pour les utilisateurs individuels). Cette couche d'autorisation s'appuie sur l'utilisation d'informations d'identification temporaires. Pour cette raison, vous ne pouvez pas utiliser de `aws:TokenIssueTime` condition pour vérifier que les utilisateurs utilisent des informations d'identification à long terme ou pour bloquer les actions des utilisateurs qui utilisent des informations d'identification temporaires, comme décrit dans la référence aux éléments de politique IAM JSON dans la documentation IAM.

Gestion des autorisations de AWS OpsWorks Stacks en joignant une politique IAM

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez spécifier les autorisations AWS OpsWorks Stacks d'un utilisateur en joignant une politique IAM. Une stratégie attachée est requise pour certaines autorisations :

- Autorisations utilisateur administratif, comme l'importation d'utilisateurs.
- Autorisations pour certaines actions, telles que la création ou le clonage d'une pile.

Pour obtenir la liste complète des actions qui nécessitent une stratégie attachée, consultez AWS OpsWorks Stabilise les niveaux d'autorisation.

Vous pouvez également utiliser une politique pour personnaliser les niveaux d'autorisation accordés via la page Autorisations. Cette section fournit un bref résumé de la manière d'appliquer une politique IAM à un utilisateur pour spécifier les autorisations AWS OpsWorks Stacks. Pour plus d'informations, consultez la section Gestion des accès aux AWS ressources.

Une politique IAM est un objet JSON qui contient une ou plusieurs instructions. Chaque élément de la déclaration possède une liste d'autorisations, composées chacune de trois éléments de base :

Action

Les actions que l'autorisation affecte. Vous spécifiez les actions AWS OpsWorks Stacks sous la formeopsworks:*action*. Une `Action` peut être définie sur une action spécifique comme `opsworks:CreateStack`, qui indique si l'utilisateur est autorisé à appeler `CreateStack`. Vous pouvez également utiliser des caractères génériques pour spécifier des groupes d'actions. Par exemple, `opsworks:Create*` spécifie toutes les actions de création. Pour une liste complète des actions AWS OpsWorks Stacks, consultez la référence de l'API AWS OpsWorks Stacks.

Effet

Indique si les actions spécifiées sont autorisées ou refusées.

Ressource

Les AWS ressources concernées par l'autorisation. AWS OpsWorks Stacks possède un seul type de ressource, la pile. Pour spécifier les autorisations d'une ressource de pile particulière, définissez `Resource` sur l'ARN de la pile, qui a le format suivant : `arn:aws:opsworks:`*region*`:`*account_id*`:stack/`*stack_id*`/`.

Vous pouvez également utiliser des caractères génériques. Par exemple, la définition de `Resource` sur `*` attribue les autorisations pour chaque ressource.

Par exemple, la stratégie suivante refuse à l'utilisateur la possibilité d'arrêter les instances sur la pile dont l'ID est `2860-2f18b4cb-4de5-4429-a149-ff7da9f0d8ee`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "opsworks:StopInstance",
      "Effect": "Deny",
      "Resource": "arn:aws:opsworks:*:*:stack/2f18b4cb-4de5-4429-a149-ff7da9f0d8ee/"
    }
  ]
}
```

Pour plus d'informations sur l'ajout d'autorisations à un utilisateur IAM, consultezhttps://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_change-permissions.html#users_change_permissions-add-console.

Pour plus d'informations sur la façon de créer ou de modifier des politiques IAM, consultez la section Politiques et autorisations dans IAM. Pour des exemples de politiques de AWS OpsWorks Stacks, voirExemples de stratégies.

Exemples de stratégies

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section décrit des exemples de politiques IAM qui peuvent être appliquées aux utilisateurs de AWS OpsWorks Stacks.

- Autorisations administrativesdécrit les politiques utilisées pour accorder des autorisations aux utilisateurs administratifs.

- Gestion des autorisationset Autorisations Deploy montrez des exemples de politiques qui peuvent être appliquées à un utilisateur pour augmenter ou restreindre les niveaux d'autorisation de gestion et de déploiement.

  AWS OpsWorks Stacks détermine les autorisations de l'utilisateur en évaluant les autorisations accordées par les politiques IAM ainsi que les autorisations accordées par la page Permissions. Pour plus d'informations, consultez la section Contrôle de l'accès aux AWS ressources à l'aide de politiques. Pour plus d'informations sur les autorisations de la page Autorisations, consultez AWS OpsWorks Stabilise les niveaux d'autorisation.

Autorisations administratives

Utilisez la console IAM https://console.aws.amazon.com/iam/, pour accéder à la AWSOpsWorks_FullAccess politique, attachez cette politique à un utilisateur pour lui accorder l'autorisation d'effectuer toutes les actions AWS OpsWorks Stacks. Les autorisations IAM sont requises, entre autres, pour permettre à un utilisateur administratif d'importer des utilisateurs.

Vous devez créer un rôle IAM qui permet à AWS OpsWorks Stacks d'agir en votre nom pour accéder à d'autres AWS ressources, telles que les instances Amazon EC2 . Vous gérez généralement cette tâche en demandant à un utilisateur administratif de créer la première pile et en laissant AWS OpsWorks Stacks créer le rôle pour vous. Vous pouvez ensuite utiliser ce rôle pour toutes les piles suivantes. Pour de plus amples informations, veuillez consulter Permettre à AWS OpsWorks Stacks d'agir en votre nom.

L'utilisateur administratif qui crée la première pile doit disposer d'autorisations pour certaines actions IAM qui ne sont pas incluses dans la AWSOpsWorks_FullAccess politique. Ajoutez les autorisations suivantes à la `Actions` section de la politique. Pour une syntaxe JSON correcte, veillez à ajouter des virgules entre les actions et à supprimer la virgule de fin à la fin de la liste des actions.

```
"iam:PutRolePolicy",
"iam:AddRoleToInstanceProfile",
"iam:CreateInstanceProfile",
"iam:CreateRole"
```

Gestion des autorisations

Le niveau d'autorisation Gérer permet à un utilisateur d'effectuer diverses actions de gestion de pile, y compris l'ajout ou la suppression de couches. Cette rubrique décrit plusieurs politiques que

vous pouvez utiliser pour gérer les utilisateurs afin d'augmenter ou de restreindre les autorisations standard.

Refuser à un utilisateur Gérer la possibilité d'ajouter ou de supprimer des couches

Vous pouvez restreindre le niveau des autorisations de gestion pour permettre à un utilisateur d'effectuer toutes les actions de gestion, à l'exception de l'ajout ou de la suppression de couches, en utilisant la politique IAM suivante. Remplacez *regionaccount_id*, et *stack_id* par des valeurs adaptées à votre configuration.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "opsworks:CreateLayer",
        "opsworks:DeleteLayer"
      ],
      "Resource": "arn:aws:opsworks:region:account_id:stack/stack_id/"
    }
  ]
}
```

Permettre à un utilisateur Gérer de créer ou de cloner des piles

Le niveau Gérer les autorisations ne permet pas aux utilisateurs de créer ou de cloner des piles. Vous pouvez modifier les autorisations de gestion pour permettre à un utilisateur de créer ou de cloner des piles en appliquant la politique IAM suivante. Remplacez *region* et *account_id* par des valeurs adaptées à votre configuration.

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

Refuser à un utilisateur Manage la possibilité d'enregistrer des ressources ou d'annuler leur enregistrement

Le niveau de gestion des autorisations permet à l'utilisateur d'enregistrer et de désenregistrer les ressources d'adresses IP Amazon EBS et Elastic auprès de la pile. Vous pouvez restreindre les autorisations de gestion pour permettre à l'utilisateur d'effectuer toutes les actions de gestion, à l'exception de l'enregistrement des ressources, en appliquant la politique suivante.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "opsworks:RegisterVolume",
        "opsworks:RegisterElasticIp"
      ],
      "Resource": "*"
    }
  ]
}
```

Autoriser un utilisateur Gérer à importer des utilisateurs

Le niveau de gestion des autorisations ne permet pas aux utilisateurs d'importer des utilisateurs dans AWS OpsWorks Stacks. Vous pouvez augmenter les autorisations de gestion pour permettre à un utilisateur d'importer et de supprimer des utilisateurs en appliquant la politique IAM suivante. Remplacez *region* et *account_id* par des valeurs adaptées à votre configuration.

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

Autorisations Deploy

Le niveau d'autorisation Déployer ne permet pas aux utilisateurs de créer ou de supprimer des applications. Vous pouvez augmenter les autorisations de déploiement pour permettre à un utilisateur de créer et de supprimer des applications en appliquant la politique IAM suivante. Remplacez *regionaccount_id*, et *stack_id* par des valeurs adaptées à votre configuration.

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

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "opsworks:CreateApp",
        "opsworks:DeleteApp"
      ],
      "Resource": "arn:aws:opsworks:region:account_id:stack/stack_id/"
    }
  ]
}
```

AWS OpsWorks Stabilise les niveaux d'autorisation

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section répertorie les actions autorisées par les niveaux d'autorisation Afficher, déployer et gérer sur la page AWS OpsWorks Stacks Permissions. Il inclut également une liste d'actions auxquelles vous ne pouvez accorder des autorisations qu'en appliquant une politique IAM à l'utilisateur.

Afficher

Le niveau Show autorise les commandes `Describe`*XYZ*, avec les exceptions suivantes :

```
DescribePermissions
DescribeUserProfiles
DescribeMyUserProfile
DescribeStackProvisioningParameters
```

Si un utilisateur administrateur a activé la gestion automatique pour l'utilisateur, les utilisateurs Show peuvent également utiliser `DescribeMyUserProfile` et `UpdateMyUserProfile`. Pour plus d'informations sur la gestion automatique, consultez Modification des paramètres utilisateur.

Déploiement

Les actions suivantes sont autorisées par le niveau Deploy, en plus des actions autorisées par le niveau Show.

```
CreateDeployment
UpdateApp
```

Gérer

Les actions suivantes sont autorisées par le niveau Manage, en plus des actions autorisées par les niveaux Deploy et Show.

```
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
```

Autorisations nécessitant une politique IAM

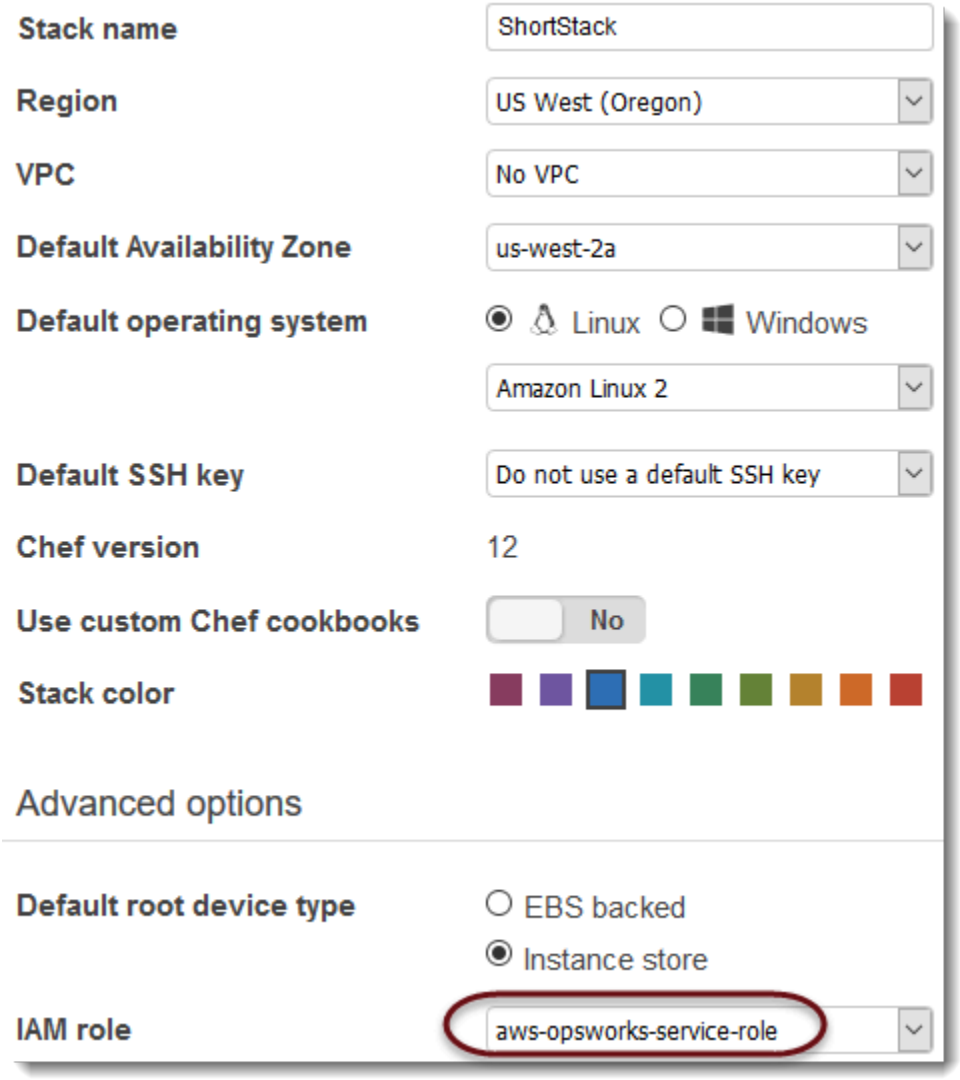
Vous devez accorder des autorisations pour les actions suivantes en appliquant une politique IAM appropriée à l'utilisateur. Pour obtenir quelques exemples, consultez Exemples de stratégies.

```
CloneStack
CreateStack
CreateUserProfile
DeleteUserProfile
DescribeUserProfiles
UpdateUserProfile
```

Permettre à AWS OpsWorks Stacks d'agir en votre nom

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks doit interagir avec divers services AWS en votre nom. Par exemple, AWS OpsWorks Stacks interagit avec Amazon EC2 pour créer des instances et avec Amazon CloudWatch pour obtenir des statistiques de surveillance. Lorsque vous créez une pile, vous spécifiez un rôle IAM, généralement appelé rôle de service, qui accorde à AWS OpsWorks Stacks les autorisations appropriées.



Lorsque vous spécifiez le rôle de service d'une nouvelle pile, vous pouvez effectuer l'une des actions suivantes :

- Spécifier un rôle de service standard que vous avez créé plus tôt.

  Vous pouvez généralement créer un rôle de service standard lorsque vous créez votre première pile, puis utiliser ce rôle pour toutes les piles suivantes.
- Spécifiez un rôle de service personnalisé que vous avez créé à l'aide de la console ou de l'API IAM.

  Cette approche est utile si vous souhaitez accorder à AWS OpsWorks Stacks des autorisations plus limitées que le rôle de service standard.

> **Note**
>
> Pour créer votre première pile, vous devez disposer des autorisations définies dans le modèle de AdministratorAccesspolitique IAM. Ces autorisations permettent à AWS OpsWorks Stacks de créer un rôle de service IAM et vous permettent d'importer des utilisateurs, comme indiqué plus tôt. Pour toutes les piles suivantes, les utilisateurs peuvent sélectionner le rôle de service créé pour la première pile ; ils n'ont pas besoin d'autorisations d'administration complètes pour créer une pile.

Le rôle de service standard octroie les autorisations suivantes :

- Effectuez toutes les EC2 actions Amazon (`ec2:*`).
- Obtenez CloudWatch des statistiques (`cloudwatch:GetMetricStatistics`).
- Utilisez Elastic Load Balancing pour distribuer le trafic vers les serveurs (`elasticloadbalancing:*`).
- Utilisez une instance Amazon RDS comme serveur de base de données (`rds:*`).
- Utilisez les rôles IAM (`iam:PassRole`) pour garantir une communication sécurisée entre AWS OpsWorks Stacks et vos instances Amazon EC2 .

Si vous créez un rôle de service personnalisé, vous devez vous assurer qu'il accorde toutes les autorisations dont AWS OpsWorks Stacks a besoin pour gérer votre stack. L'exemple JSON suivant est la déclaration de stratégie pour le rôle de service standard ; un rôle de service personnalisé doit au moins inclure les autorisations suivantes dans sa déclaration de stratégie.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:*",
                "iam:PassRole",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:DescribeAlarms",
                "ecs:*",
                "elasticloadbalancing:*",
                "rds:*"
            ],
```

```
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ec2.amazonaws.com"
                }
            }
        }
    ]
}
```

Un rôle de service a également une relation d'approbation. Les rôles de service créés par AWS OpsWorks Stacks ont la relation de confiance suivante.

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

Le rôle de service doit avoir cette relation de confiance pour que AWS OpsWorks Stacks puisse agir en votre nom. Si vous utilisez le rôle de service par défaut, ne modifiez pas la relation d'approbation. Si vous créez un rôle de service personnalisé, spécifiez la relation de confiance en effectuant l'une des opérations suivantes :

- Si vous utilisez l'assistant de création de rôle dans la console IAM, dans Choisir un cas d'utilisation, choisissez Opsworks. Ce rôle possède la relation de confiance appropriée, mais aucune politique n'y est implicitement attachée. Pour autoriser AWS OpsWorks Stacks à agir en votre nom, créez une politique gérée par le client contenant les éléments suivants et associez-la au nouveau rôle.

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

- Si vous utilisez un AWS CloudFormation modèle, vous pouvez ajouter quelque chose comme ce qui suit à la section Ressources de votre modèle.

```
"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": {
              ...
            } ]
        }
      },
   }
}
```

Prévention interservices confuse des adjoints dans Stacks AWS OpsWorks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner un problème de confusion chez les adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé d'accéder

autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés de contexte de condition `aws:SourceAccount`globale `aws:SourceArn`et les clés de contexte dans les politiques d'accès aux piles afin de limiter les autorisations que AWS OpsWorks Stacks accorde à un autre service aux piles. Si la valeur `aws:SourceArn` ne contient pas l'ID du compte, tel qu'un ARN de compartiment Amazon S3, vous devez utiliser les deux clés de contexte de condition globale pour limiter les autorisations. Si vous utilisez les deux clés de contexte de condition globale et que la valeur `aws:SourceArn` contient l'ID de compte, la valeur `aws:SourceAccount` et le compte dans la valeur `aws:SourceArn` doivent utiliser le même ID de compte lorsqu'ils sont utilisés dans la même instruction de politique. À utiliser `aws:SourceArn` si vous souhaitez qu'une seule pile soit associée à l'accès multiservice. À utiliser `aws:SourceAccount` si vous souhaitez autoriser toute pile de ce compte à être associée à l'utilisation interservices.

La valeur de `aws:SourceArn` doit être l'ARN d'une AWS OpsWorks pile.

Le moyen le plus efficace de se protéger contre le problème de confusion des adjoints consiste à utiliser la clé de contexte de condition `aws:SourceArn` globale avec l'ARN complet de la pile AWS OpsWorks Stacks. Si vous ne connaissez pas l'ARN complet ou si vous spécifiez plusieurs piles ARNs, utilisez la clé de condition de contexte `aws:SourceArn` globale avec des caractères génériques (*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

La section suivante explique comment utiliser les clés contextuelles `aws:SourceArn` et les clés de contexte de condition `aws:SourceAccount` globale dans AWS OpsWorks Stacks pour éviter le problème de confusion des adjoints.

Empêchez les exploits secondaires confus dans AWS OpsWorks Stacks

Cette section décrit comment vous pouvez empêcher les exploits secondaires confus dans AWS OpsWorks Stacks et inclut des exemples de politiques d'autorisation que vous pouvez associer au rôle IAM que vous utilisez pour accéder AWS OpsWorks à Stacks. Pour des raisons de sécurité, nous vous recommandons d'ajouter les clés de `aws:SourceAccount` condition `aws:SourceArn` et aux relations de confiance que votre rôle IAM entretient avec d'autres services. Les relations de confiance permettent à AWS OpsWorks Stacks d'assumer un rôle pour effectuer des actions dans d'autres services nécessaires à la création ou à la gestion de vos AWS OpsWorks stacks Stacks.

Pour modifier les relations de confiance afin d'ajouter **`aws:SourceArn`** et de **`aws:SourceAccount`** conditionner des clés

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation de gauche, choisissez Rôles.
3. Dans la zone de recherche, recherchez le rôle que vous utilisez pour accéder à AWS OpsWorks Stacks. Le rôle AWS géré est`aws-opsworks-service-role`.
4. Sur la page Résumé du rôle, choisissez l'onglet Relations de confiance.
5. Dans l'onglet Relations de confiance, choisissez Modifier la politique de confiance.
6. Sur la page Modifier la politique de confiance, ajoutez au moins l'une des clés de `aws:SourceAccount` condition `aws:SourceArn` ou à la politique. `aws:SourceArn`À utiliser pour restreindre la relation de confiance entre les services croisés (tels qu'Amazon EC2) et AWS OpsWorks Stacks à des piles AWS OpsWorks Stacks spécifiques, ce qui est plus restrictif. Ajoutez `aws:SourceAccount` pour limiter la relation de confiance entre cross services et AWS OpsWorks Stacks aux piles d'un compte spécifique, ce qui est moins restrictif. Voici un exemple. Notez que si vous utilisez les deux clés de condition, le compte IDs doit être le même.

```
{
  "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. Lorsque vous avez terminé d'ajouter des clés de condition, choisissez Mettre à jour la politique.

Vous trouverez ci-dessous d'autres exemples de rôles qui limitent l'accès aux piles en utilisant `aws:SourceArn` et`aws:SourceAccount`.

Rubriques

- Exemple : accès aux piles dans une région spécifique
- Exemple : ajout de plusieurs ARN de pile à aws:SourceArn

Exemple : accès aux piles dans une région spécifique

La déclaration de relation de confiance suivante permet d'accéder à toutes les AWS OpsWorks piles Stacks de la région USA Est (Ohio) (). `us-east-2` Notez que la région est spécifiée dans la valeur ARN de`aws:SourceArn`, mais que la valeur de l'ID de pile est un caractère générique (*).

```
{
  "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/*"
        }
      }
    }
  ]
}
```

Exemple : ajout de plusieurs ARN de pile à **`aws:SourceArn`**

L'exemple suivant limite l'accès à un tableau de deux AWS OpsWorks piles Stacks dans l'ID de compte 123456789012.

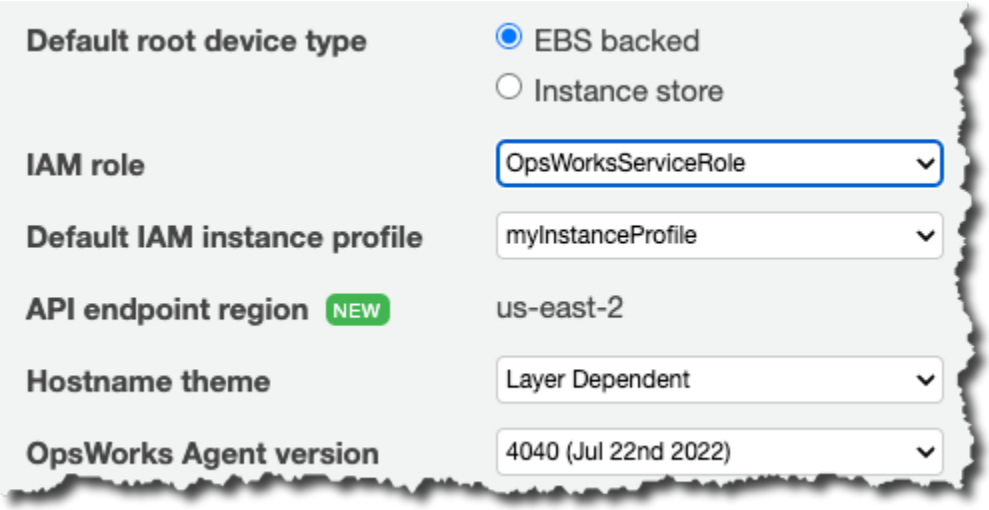
```
{
```

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

Spécification des autorisations pour les applications exécutées sur EC2 des instances

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Si les applications exécutées sur les EC2 instances Amazon de votre stack doivent accéder à d'autres ressources AWS, telles que les compartiments Amazon S3, elles doivent disposer des autorisations appropriées. Pour attribuer ces autorisations, vous utilisez un profil d'instance. Vous pouvez spécifier un profil d'instance pour chaque instance lorsque vous créez une pile AWS OpsWorks Stacks.



Vous pouvez également spécifier un profil pour les instances d'une couche en modifiant la configuration de la couche.

Le profil d'instance spécifie un rôle IAM. Les applications qui s'exécutent sur l'instance peuvent assumer ce rôle pour accéder aux ressources AWS, sous réserve des autorisations accordées par la stratégie du rôle. Pour plus d'informations sur la façon dont une application assume un rôle, consultez Assumer un rôle à l'aide d'un appel d'API.

Vous pouvez créer un profil d'instance de différentes façons :

- Utilisez la console ou l'API IAM pour créer un profil.

  Pour plus d'informations, consultez Rôles (délégation et fédération).
- Utilisez un AWS CloudFormation modèle pour créer un profil.

  Pour des exemples d'inclusion de ressources IAM dans un modèle, consultez Identity and Access Management (IAM) Template Snippets.

Un profil d'instance doit avoir une relation d'approbation et une stratégie attachée qui accorde les autorisations d'accès aux ressources AWS.

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

```
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Le profil d'instance doit avoir cette relation de confiance pour que AWS OpsWorks Stacks puisse agir en votre nom. Si vous utilisez le rôle de service par défaut, ne modifiez pas la relation d'approbation. Si vous créez un rôle de service personnalisé, spécifiez la relation d'approbation comme suit :

- Si vous utilisez l'assistant de création de rôle dans la console IAM, spécifiez le type de EC2 rôle Amazon sous AWS Service Roles sur la deuxième page de l'assistant.
- Si vous utilisez un AWS CloudFormation modèle, vous pouvez ajouter quelque chose comme ce qui suit à la section Ressources de votre modèle.

```
"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"
            }
         ]
      }
```

```
    }
}
```

Lorsque vous créez votre profil d'instance, vous pouvez associer une politique appropriée au rôle du profil à ce moment-là. Après avoir créé la pile, vous devez utiliser la console ou l'API IAM pour associer une politique appropriée au rôle du profil. Par exemple, la politique suivante accorde un accès complet à tous les objets du compartiment Amazon S3 nommé amzn-s3-demo-bucket. Remplacez *region* et amzn-s3-demo-bucket par des valeurs adaptées à votre configuration.

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

Pour obtenir un exemple de la façon de créer et d'utiliser un profil d'instance, consultez Utilisation d'un compartiment Amazon S3.

Si votre application utilise un profil d'instance pour appeler l'API AWS OpsWorks Stacks depuis une EC2 instance, la politique doit autoriser l'`iam:PassRole`action en plus des actions appropriées pour AWS OpsWorks Stacks et les autres services AWS. L'autorisation `iam:PassRole` permet à AWS OpsWorks Stacks d'assumer le rôle de service en votre nom. Pour plus d'informations sur l'API AWS OpsWorks Stacks, consultez le manuel de référence des OpsWorks API AWS.

Voici un exemple de politique IAM qui vous permet d'appeler n'importe quelle action AWS OpsWorks Stacks depuis une EC2 instance, ainsi que toute action Amazon EC2 ou Amazon S3.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:*",
        "s3:*",
        "opsworks:*",
```

```
        "iam:PassRole"
      ],
      "Resource": "arn:aws:ec2:region:account_id:instance/*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "opsworks.amazonaws.com"
        }
      }
    }
  ]
}
```

> **Note**
>
> Si vous ne l'autorisez pas`iam:PassRole`, toute tentative d'appel d'une action AWS OpsWorks Stacks échoue avec une erreur comme celle-ci :
>
> ```
> User: arn:aws:sts::123456789012:federated-user/Bob is not authorized
> to perform: iam:PassRole on resource:
> arn:aws:sts::123456789012:role/OpsWorksStackIamRole
> ```

Pour plus d'informations sur l'utilisation des rôles sur une EC2 instance pour obtenir des autorisations, consultez la section Accorder aux applications exécutées sur EC2 des instances Amazon l'accès aux ressources AWS dans le guide de AWS Identity and Access Management l'utilisateur.

Gestion de l'accès SSH

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks prend en charge les clés SSH pour les piles Linux et Windows.

- Pour les instances Linux, vous pouvez utiliser SSH pour vous connecter à une instance : par exemple, pour exécuter les commandes de l'interface de ligne de commande de l'agent.

  Pour de plus amples informations, veuillez consulter Connexion avec SSH.
- Pour les instances Windows, vous pouvez utiliser une clé SSH pour obtenir le mot de passe Administrateur de l'instance, que vous pouvez ensuite utiliser pour vous connecter avec le protocole RDP.

  Pour de plus amples informations, veuillez consulter Connexion avec RDP.

L'authentification est basée sur une paire de clés SSH, qui se compose d'une clé publique et d'une clé privée :

- Vous installez la clé publique sur l'instance.

  L'emplacement dépend du système d'exploitation en question, mais AWS OpsWorks Stacks s'occupe des détails pour vous.
- Vous stockez la clé privée localement et la fournissez à un client SSH, tel que `ssh.exe`, pour accéder à l'instance.

  Le client SSH utilise la clé privée pour se connecter à l'instance.

Pour fournir un accès SSH aux utilisateurs d'une pile, vous avez besoin d'un moyen de créer les paires de clés SSH, d'installer les clés publiques sur les instances de la pile et de gérer en toute sécurité les clés privées.

Amazon EC2 propose un moyen simple d'installer une clé SSH publique sur une instance. Vous pouvez utiliser la EC2 console ou l'API Amazon pour créer une ou plusieurs paires de clés pour chaque région AWS que vous prévoyez d'utiliser. Amazon EC2 stocke les clés publiques sur AWS et vous stockez les clés privées localement. Lorsque vous lancez une instance, vous spécifiez l'une des paires de clés de la région et Amazon l'installe EC2 automatiquement sur l'instance. Vous utilisez ensuite la clé privée correspondante pour vous connecter à l'instance. Pour plus d'informations, consultez Amazon EC2 Key Pairs.

Avec AWS OpsWorks Stacks, vous pouvez spécifier l'une des paires de EC2 clés Amazon de la région lorsque vous créez une pile, et éventuellement la remplacer par une paire de clés différente lorsque vous créez chaque instance. Lorsque AWS OpsWorks Stacks lance l' EC2 instance Amazon

correspondante, il spécifie la paire de clés et Amazon EC2 installe la clé publique sur l'instance. Vous pouvez ensuite utiliser la clé privée pour vous connecter ou récupérer un mot de passe administrateur, comme vous le feriez avec une EC2 instance Amazon standard. Pour de plus amples informations, veuillez consulter Installation d'une EC2 clé Amazon.

L'utilisation d'une paire de EC2 clés Amazon est pratique, mais présente deux limites importantes :

- Une paire de EC2 clés Amazon est liée à une région AWS spécifique.

  Si vous travaillez dans plusieurs régions, vous devez gérer plusieurs paires de clés.
- Vous ne pouvez installer qu'une seule paire de EC2 clés Amazon sur une instance.

  Si vous voulez autoriser plusieurs utilisateurs à se connecter, ils doivent tous avoir une copie de la clé privée, ce qui n'est pas une méthode de sécurité recommandée.

Pour les piles Linux, AWS OpsWorks Stacks fournit un moyen plus simple et plus flexible de gérer les paires de clés SSH.

- Chaque utilisateur enregistre une paire de clés personnelle.

  Ils stockent la clé privée localement et enregistrent la clé publique auprès de AWS OpsWorks Stacks, comme décrit dansEnregistrement de la clé SSH publique d'un utilisateur.
- Lorsque vous définissez les autorisations utilisateur pour une pile, vous spécifiez les utilisateurs qui doivent bénéficier d'un accès SSH aux instances de la pile.

  AWS OpsWorks Stacks crée automatiquement un utilisateur système sur les instances de la pile pour chaque utilisateur autorisé et installe sa clé publique. L'utilisateur peut ensuite utiliser la clé privée correspondante pour se connecter, comme décrit dans Connexion avec SSH.

L'utilisation de clés SSH personnelles présente les avantages suivants.

- Il n'est pas nécessaire de configurer manuellement les clés sur les instances ; AWS OpsWorks Stacks installe automatiquement les clés publiques appropriées sur chaque instance.
- AWS OpsWorks Stacks installe uniquement les clés publiques personnelles des utilisateurs autorisés.

Les utilisateurs non autorisés ne peuvent pas utiliser leur clé privée personnelle pour accéder aux instances. Avec les paires de EC2 clés Amazon, tout utilisateur possédant la clé privée correspondante peut se connecter, avec ou sans accès SSH autorisé.

- Si un utilisateur n'a plus besoin de l'accès SSH, vous pouvez utiliser la page Autorisations pour révoquer les autorisations SSH/RDP de l'utilisateur.

  AWS OpsWorks Stacks désinstalle immédiatement la clé publique des instances de la pile.
- Vous pouvez utiliser la même clé pour n'importe quelle région AWS.

  Les utilisateurs ne doivent gérer qu'une seule clé privée.
- Il n'y a pas besoin de partager les clés privées.

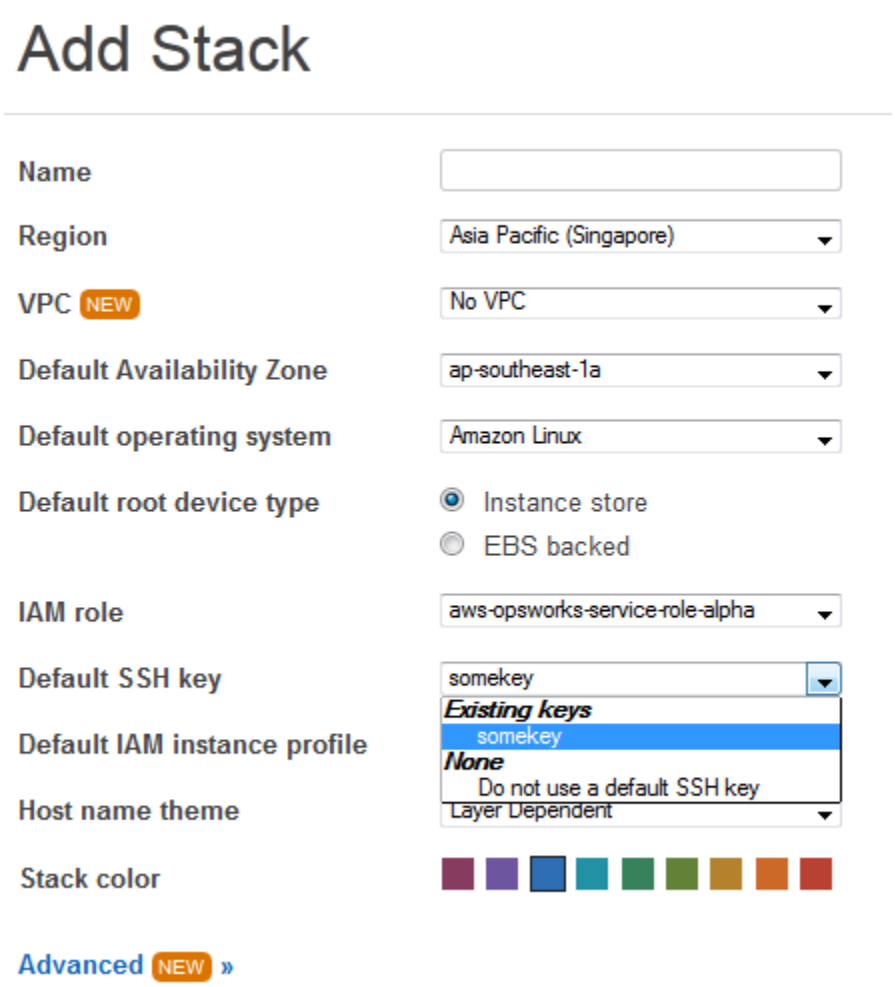
  Chaque utilisateur a sa propre clé privée.
- Il est facile d'effectuer une rotation des clés.

  Vous ou l'utilisateur mettez à jour la clé publique dans Mes paramètres et AWS OpsWorks Stacks met automatiquement à jour les instances.

Installation d'une EC2 clé Amazon

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

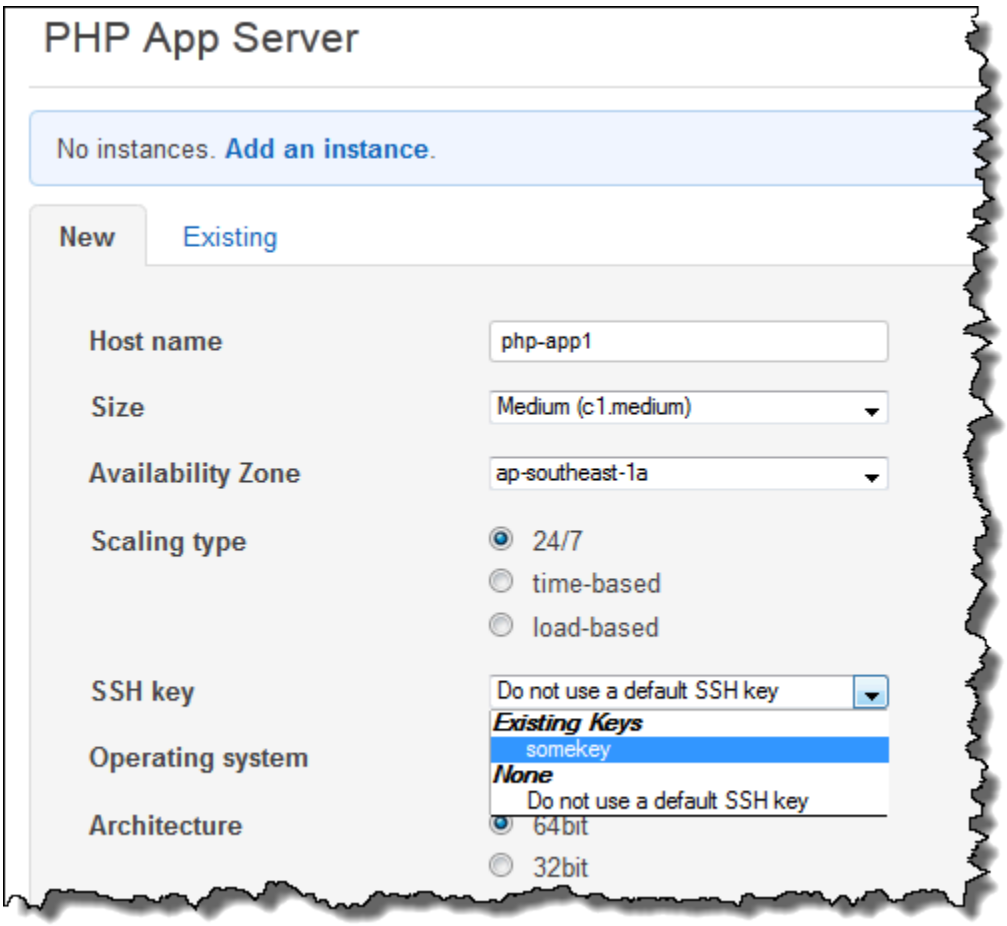
Lorsque vous créez une pile, vous pouvez spécifier une clé Amazon EC2 SSH installée par défaut sur chaque instance de la pile.



La liste des clés SSH par défaut indique Amazon EC2keys de votre compte AWS. Vous pouvez effectuer l'une des actions suivantes :

- Sélectionnez la clé appropriée dans la liste.
- Sélectionnez Ne pas utiliser de clé SSH par défaut pour ne spécifier aucun clé.

Si vous avez sélectionné Ne pas utiliser de clé SSH par défaut ou que vous ne voulez pas remplacer la clé par défaut d'une pile, vous pouvez spécifier une clé lorsque vous créez une instance.



Lorsque vous démarrez l'instance, AWS OpsWorks Stacks installe la clé publique dans le `authorized_keys` fichier.

Enregistrement de la clé SSH publique d'un utilisateur

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Il existe deux façons d'enregistrer la clé publique SSH d'un utilisateur :

- Un utilisateur administratif peut attribuer une clé publique SSH à un ou plusieurs utilisateurs et leur offrir la clé privée correspondante.
- Un utilisateur administratif peut activer la gestion automatique pour un ou plusieurs utilisateurs.

  Ces utilisateurs peuvent alors spécifier leur propre clé publique SSH.

Pour plus d'informations sur la façon dont les utilisateurs administratifs peuvent activer la gestion automatique ou attribuer des clés publiques aux utilisateurs, consultez Modification des paramètres utilisateur.

La connexion à des instances basées sur Linux à l'aide de SSH dans un terminal PuTTY requiert des étapes supplémentaires. Pour plus d'informations, consultez la page Connexion à votre instance Linux à partir de Windows à l'aide de PuTTY et Dépannage de la connexion à votre instance dans la documentation AWS.

Ce qui suit décrit comment un utilisateur dont l'autogestion est activée peut spécifier sa clé publique.

Pour spécifier votre clé publique SSH

1. Créez une paire de clés SSH.

   L'approche la plus simple consiste à générer la paire de clés localement. Pour plus d'informations, consultez Comment générer votre propre clé et l'importer sur Amazon EC2.

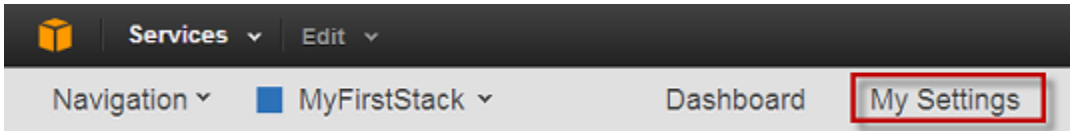
   > **Note**
   >
   > Si vous utilisez Pu TTYgen pour générer votre paire de clés, copiez la clé publique depuis la clé publique pour la coller dans le champ de fichier openSSH authorized_keys. Cliquez sur Enregistrer la clé publique pour enregistrer la clé publique dans un format non pris en charge par MindTerm.

2. Connectez-vous à la console AWS OpsWorks Stacks en tant qu'utilisateur IAM avec l'autogestion activée.

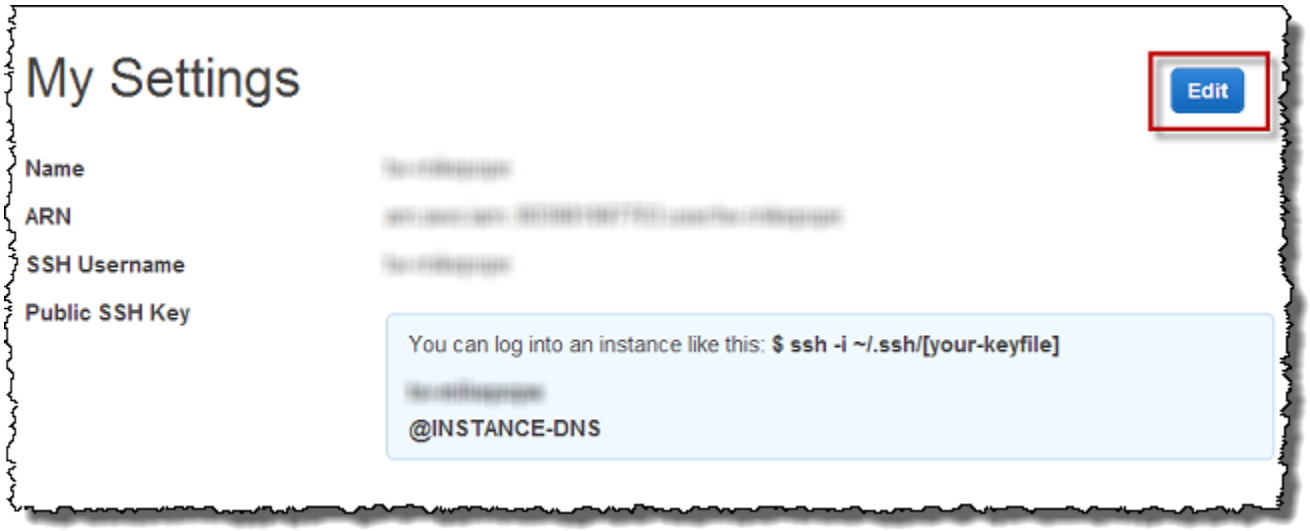
   > **Important**
   >
   > Si vous vous connectez en tant que propriétaire du compte ou en tant qu'utilisateur IAM pour lequel l'autogestion n'est pas activée, AWS OpsWorks Stacks n'affiche pas Mes paramètres. Si vous êtes un utilisateur administratif ou le propriétaire du compte, vous

pouvez spécifier à la place les clés SSH en accédant à la page Users et en modifiant les paramètres de l'utilisateur.

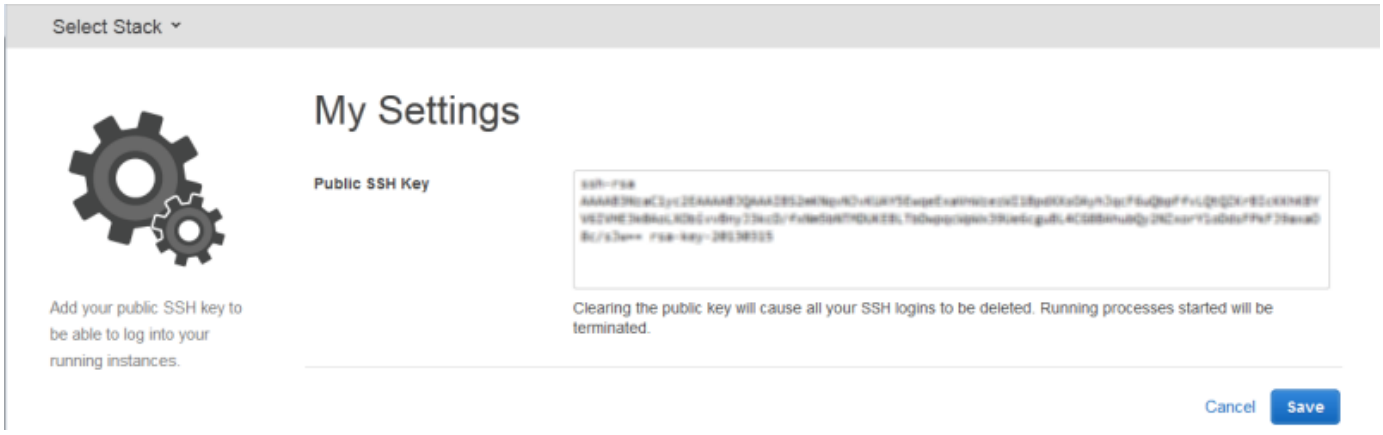
3. Sélectionnez Mes paramètres, qui affiche les paramètres de l'utilisateur connecté.



4. Sur la page Mes paramètres, cliquez sur Modifier.



5. Dans la zone Clé publique SSH, saisissez votre clé publique SSH, puis cliquez sur Enregistrer.



Important

Pour utiliser le client MindTerm SSH intégré pour se connecter aux EC2 instances Amazon, un utilisateur doit être connecté en tant qu'utilisateur IAM et disposer d'une clé SSH publique

enregistrée auprès de Stacks. AWS OpsWorks Pour de plus amples informations, veuillez consulter Utilisation du client MindTerm SSH intégré.

Gestion des mises à jour de sécurité Linux

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Mises à jour de sécurité

Les fournisseurs de systèmes d'exploitation Linux fournissent des mises à jour régulières, dont la plupart sont des correctifs de sécurité du système d'exploitation, mais elles peuvent aussi inclure les mises à jour des packages installés. Vous devez vous assurer que les systèmes d'exploitation de vos instances sont à jour avec les derniers correctifs de sécurité.

Par défaut, AWS OpsWorks Stacks installe automatiquement les dernières mises à jour lors de l'installation, une fois le démarrage d'une instance terminé. AWS OpsWorks Stacks n'installe pas automatiquement les mises à jour une fois qu'une instance est en ligne, afin d'éviter des interruptions telles que le redémarrage des serveurs d'applications. Au lieu de cela, comme vous gérez les mises à jour de vos instances en ligne vous-même, vous pouvez réduire les perturbations.

Nous vous recommandons d'utiliser l'une des actions suivantes pour mettre à jour vos instances en ligne.

- Créez et démarrez de nouvelles instances pour remplacer vos instances en ligne actuelles. Puis, supprimez les instances en cours.

  Les nouvelles instances bénéficient de l'ensemble des correctifs de sécurité les plus récents pendant l'installation.
- Sur les instances Linux des piles Chef 11.10 ou plus anciennes, exécutez la commande de pile Mettre à jour les dépendances, qui installe le jeu actuel de correctifs de sécurité et autres mises à jour sur les instances spécifiées.

Pour ces deux approches, AWS OpsWorks Stacks effectue la mise à jour en l'exécutant `yum update` pour Amazon Linux et Red Hat Enterprise Linux (RHEL) ou `apt-get update` pour Ubuntu. Comme chaque distribution gère les mises à jour un peu différemment, vous devez examiner les informations contenues dans les liens associés pour comprendre exactement comment une mise à jour affecte vos instances :

- Amazon Linux — Les mises à jour d'Amazon Linux installent des correctifs de sécurité et peuvent également installer des mises à jour de fonctionnalités, notamment des mises à jour de packages.

  Pour plus d'informations, consultez AMI Amazon Linux FAQs.
- Ubuntu — Les mises à jour d'Ubuntu se limitent en grande partie à l'installation de correctifs de sécurité, mais peuvent également installer des mises à jour de packages pour un nombre limité de correctifs critiques.

  Pour plus d'informations, consultez LTS - Ubuntu Wiki.
- CentOS — Les mises à jour de CentOS conservent généralement la compatibilité binaire avec les versions antérieures.
- RHEL — Les mises à jour de RHEL maintiennent généralement la compatibilité binaire avec les versions antérieures.

  Pour plus d'informations, consultez Cycle de vie Red Hat Enterprise Linux.

Si vous souhaitez mieux contrôler les mises à jour, par exemple en spécifiant des versions de package spécifiques, vous pouvez désactiver les mises à jour automatiques en utilisant les UpdateLayeractions CreateInstance, UpdateInstance, ou CreateLayer, ou les méthodes du SDK AWS ou les commandes de la CLI AWS équivalentes, pour définir le paramètre sur. `InstallUpdatesOnBoot` `false` L'exemple suivant montre comment utiliser l'interface de ligne de commande AWS pour désactiver `InstallUpdatesOnBoot` comme paramètre par défaut d'une couche existante.

```
aws opsworks update-layer --layer-id layer ID --no-install-updates-on-boot
```

Vous devez ensuite gérer vous-même les mises à jour. Par exemple, vous pourriez faire appel à l'une de ces stratégies :

- Mettre en œuvre une recette personnalisée qui exécute la commande shell appropriée pour installer vos mises à jour préférées.

Comme les mises à jour système ne correspondent pas naturellement à un événement du cycle de vie, incluez la recette dans vos livres de recettes personnalisés, mais exécutez-la manuellement. Pour les mises à jour de package, vous pouvez aussi utiliser les ressources yum_package (Amazon Linux) ou apt_package (Ubuntu) au lieu d'une commande shell.

- Connectez-vous à chaque instance avec SSH et exécutez les commandes appropriées manuellement.

Utilisation des groupes de sécurité

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Groupes de sécurité

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Chaque EC2 instance Amazon possède un ou plusieurs groupes de sécurité associés qui régissent le trafic réseau de l'instance, un peu comme un pare-feu. Un groupe de sécurité a une ou plusieurs règles, chacune spécifiant une catégorie spécifique de trafic autorisé. Une règle spécifie les éléments suivants :

- Le type de trafic autorisé, par exemple SSH ou HTTP
- Le protocole du trafic, tel que TCP ou UDP

- La plage d'adresses IP dont le trafic peut provenir
- La plage de ports autorisée pour le trafic

Les groupes de sécurité ont deux types de règles :

- Les règles de trafic entrant qui gèrent le trafic réseau entrant.

  Par exemple, les instances de serveur d'applications ont généralement une règle entrante qui autorise le trafic HTTP entrant à partir de n'importe quelle adresse IP sur le port 80 et une autre règle de trafic entrant qui autorise le trafic SSH entrant vers le port 22 et provenant d'une plage d'adresses IP spécifiée.
- Les règles de trafic sortant gèrent le trafic réseau sortant.

  Une pratique courante consiste à utiliser le paramètre par défaut, qui autorise tout le trafic sortant.

Pour plus d'informations sur les groupes de sécurité, consultez Amazon EC2 Security Groups.

La première fois que vous créez une pile dans une région, AWS OpsWorks Stacks crée un groupe de sécurité intégré pour chaque couche avec un ensemble de règles approprié. Tous les groupes ont des règles de trafic sortant par défaut qui autorisent tout le trafic sortant. En général, les règles de trafic entrant autorisent les éléments suivants :

- Trafic TCP, UDP et ICMP entrant provenant des couches Stacks appropriées AWS OpsWorks
- Trafic TCP entrant sur le port 22 (connexion SSH)

> **Warning**
>
> La configuration du groupe de sécurité par défaut ouvre SSH (port 22) vers n'importe quel emplacement réseau (0.0.0.0/0.). Cela permet à toutes les adresses IP d'accéder à votre instance à l'aide de SSH. Pour les environnements de production, vous devez utiliser une configuration qui autorise uniquement l'accès SSH à partir d'une adresse IP spécifique ou d'une plage d'adresses. Mettez à jour les groupes de sécurité par défaut immédiatement après leur création ou utilisez des groupes de sécurité personnalisés à la place.

- Pour les couches de serveur web, tout le trafic entrant TCP et UDP vers les ports 80 (HTTP) et 443 (HTTPS)

> **Note**
>
> Le groupe de sécurité intégré `AWS-OpsWorks-RDP-Server` est attribué à toutes les instances Windows pour autoriser un accès RDP. Toutefois, par défaut, il n'a pas de règles. Si vous exécutez une pile Windows et que vous souhaitez utiliser RDP pour accéder aux instances, vous devez ajouter une règle entrante qui autorise l'accès RDP. Pour de plus amples informations, veuillez consulter Connexion avec RDP.

Pour consulter les détails de chaque groupe, accédez à la EC2console Amazon, sélectionnez Security Groups dans le volet de navigation, puis sélectionnez le groupe de sécurité de la couche appropriée. Par exemple, AWS- OpsWorks -Default-Server est le groupe de sécurité intégré par défaut pour toutes les piles, et AWS OpsWorks WebApp - est le groupe de sécurité intégré par défaut pour la pile d'échantillons Chef 12.

> **Note**
>
> Si vous supprimez accidentellement un groupe de sécurité AWS OpsWorks Stacks, il est préférable de le recréer en demandant à AWS OpsWorks Stacks d'effectuer la tâche à votre place. Créez simplement une nouvelle pile dans la même région AWS (et un VPC, le cas échéant AWS OpsWorks). Stacks recréera automatiquement tous les groupes de sécurité intégrés, y compris celui que vous avez supprimé. Vous pouvez ensuite supprimer la pile si vous n'en avez plus l'utilisation ; les groupes de sécurité demeureront. Si vous souhaitez recréer le groupe de sécurité manuellement, il doit être une copie exacte de l'original, y compris les majuscules du nom du groupe.
> En outre, AWS OpsWorks Stacks tentera de recréer tous les groupes de sécurité intégrés si l'une des situations suivantes se produit :
>
> - Vous pouvez apporter des modifications à la page des paramètres de la pile dans la console AWS OpsWorks Stacks.
> - Vous démarrez l'un des instances de la pile.
> - Vous créez une pile.

Vous pouvez utiliser l'une des approches suivantes pour spécifier les groupes de sécurité. Vous utilisez le paramètre Utiliser les groupes de OpsWorks sécurité pour définir vos préférences lorsque vous créez une pile.

- Oui (paramètre par défaut) — AWS OpsWorks Stacks associe automatiquement le groupe de sécurité intégré approprié à chaque couche.

  Vous pouvez ajuster le groupe de sécurité intégré d'une couche en ajoutant un groupe de sécurité personnalisé avec vos paramètres préférés. Toutefois, lorsqu'Amazon EC2 évalue plusieurs groupes de sécurité, il utilise les règles les moins restrictives. Vous ne pouvez donc pas utiliser cette approche pour définir des règles plus restrictives que le groupe intégré.
- Non, AWS OpsWorks Stacks n'associe pas les groupes de sécurité intégrés aux couches.

  Vous devez créer des groupes de sécurité appropriés et en associer au moins un à chaque couche que vous créez. Cette approche permet de spécifier des règles plus contraignantes que celles des groupes intégrés. Notez qu'il reste possible d'associer manuellement un groupe de sécurité intégré à une couche. Les groupes de sécurité personnalisés ne sont nécessaires que pour les couches requérant des paramètres personnalisés.

> **Important**
>
> Si vous utilisez les groupes de sécurité intégrés, vous ne pouvez pas créer de règles plus contraignantes en modifiant manuellement les paramètres du groupe. Chaque fois que vous créez une pile, AWS OpsWorks Stacks remplace les configurations des groupes de sécurité intégrés, de sorte que toutes les modifications que vous apportez seront perdues lors de la prochaine création d'une pile. Si une couche nécessite des paramètres de groupe de sécurité plus restrictifs que le groupe de sécurité intégré, définissez Utiliser les groupes de OpsWorks sécurité sur Non, créez des groupes de sécurité personnalisés avec vos paramètres préférés et attribuez-les aux couches lors de leur création.

AWS OpsWorks Support Stacks pour Chef 12 Linux

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section fournit un bref aperçu de AWS OpsWorks Stacks pour Chef 12 Linux. Pour plus d'informations sur Chef 12 sous Windows, consultez Mise en route : Windows. Pour plus d'informations sur les versions antérieures de Chef sous Linux, consultez Chef 11.10 et versions antérieures pour Linux.

Présentation

AWS OpsWorks Stacks prend en charge Chef 12, la dernière version de Chef, pour les stacks Linux. Pour plus d'informations, consultez Découvrir Chef.

AWS OpsWorks Stacks continue de prendre en charge Chef 11.10 pour les stacks Linux. Toutefois, si vous êtes un utilisateur expérimenté de Chef qui souhaite bénéficier de la grande sélection de livres de recettes de la communauté ou écrire vos propres livres de recettes personnalisés, nous vous recommandons d'utiliser Chef 12. Les piles de Chef 12 offrent les avantages suivants par rapport à Chef 11.10 et aux piles antérieures pour Linux :

- Deux exécutions Chef distinctes : lorsqu'une commande est exécutée sur une instance, l'agent AWS OpsWorks Stacks exécute désormais deux exécutions Chef isolées : une pour les tâches qui intègrent l'instance à d'autres services AWS tels que AWS Identity and Access Management (IAM), et une pour vos livres de recettes personnalisés. La première exécution de Chef installe l'agent AWS OpsWorks Stacks sur l'instance et exécute des tâches système telles que la configuration et la gestion des utilisateurs, l'installation et la configuration des volumes, la configuration des CloudWatch métriques, etc. La deuxième exécution est dédiée exclusivement à l'exécution de vos recettes personnalisées pour AWS OpsWorks Événements de cycle de vie. Cette deuxième exécution vous permet d'utiliser vos propres livres de recettes ou ceux de la communauté de Chef.
- Résolution des conflits d'espace de noms – Avant Chef 12, AWS OpsWorks Stacks effectuait les tâches système et exécutait des recettes intégrées et personnalisées dans un environnement partagé. Cela a entraîné des conflits d'espaces de noms et un manque de clarté quant aux recettes que AWS OpsWorks Stacks avait exécutées. Les configurations par défaut indésirables ont dû être remplacées manuellement, une tâche longue et source d'erreurs. Dans Chef 12 pour Linux, AWS OpsWorks Stacks ne prend plus en charge les livres de recettes Chef intégrés pour les environnements de serveurs d'applications tels que PHP, Node.js ou Rails. En éliminant les recettes intégrées, AWS OpsWorks Stacks élimine le problème des collisions de noms entre les recettes intégrées et personnalisées.
- Support solide pour les livres de cuisine communautaires Chef — AWS OpsWorks Stacks Chef 12 Linux offre une compatibilité et un support accrus pour les livres de cuisine communautaires du supermarché Chef. Vous pouvez désormais utiliser des livres de recettes communautaires

supérieurs aux livres de recettes intégrés fournis précédemment par AWS OpsWorks Stacks, des livres de recettes conçus pour être utilisés avec les derniers environnements et frameworks de serveurs d'applications. Vous pouvez exécuter la plupart de ces livres de recettes sans modification sur Chef 12 pour Linux. Pour plus d'informations, rendez-vous sur Chef Supermarket sur le site Web de Learn Chef, le site Web de Chef Supermarket et le référentiel de livres de cuisine Chef sur GitHub.

- Mises à jour régulières de Chef 12 - AWS OpsWorks Stacks mettra à jour son environnement Chef vers la dernière version de Chef 12 peu après chaque sortie de Chef. Avec Chef 12, les mises à jour mineures de Chef et les nouvelles versions de l'agent AWS OpsWorks Stacks coïncideront. Cela vous permet de tester des nouvelles versions de Chef directement et donne la possibilité à vos recettes et applications Chef de tirer parti des dernières fonctionnalités de Chef.

Pour plus d'informations sur les versions de Chef prises en charge avant Chef 12, consultez Chef 11.10 et versions antérieures pour Linux.

Migration vers Chef 12

Les AWS OpsWorks principales modifications apportées à Chef 12 Linux par rapport à la prise en charge des versions 11.10, 11.4 et 0.9 précédentes de Chef sont les suivantes :

- Les couches intégrées ne sont plus fournies ni prises en charge pour les piles Chef 12 pour Linux. Etant donné que seules vos recettes personnalisées sont exécutées, la suppression de cette prise en charge donne une transparence totale à la mise en place de l'instance et facilite nettement l'écriture et la gestion des livres de recettes personnalisés. Par exemple, il n'est plus nécessaire de remplacer les attributs des recettes AWS OpsWorks Stacks intégrées. La suppression des couches intégrées permet également à AWS OpsWorks Stacks de mieux prendre en charge les livres de recettes développés et gérés par la communauté Chef, afin que vous puissiez en tirer pleinement parti. Les types de couches intégrés qui ne sont plus disponibles dans Chef 12 pour Linux sont : AWS Flow (Ruby), Ganglia HAProxy, Java App Server, Memcached, MySQL, Node.js App Server, PHP App Server, Rails App Server et Static Web Server.
  - Comme AWS OpsWorks Stacks exécute les recettes que vous fournissez, il n'est plus nécessaire de remplacer les attributs AWS OpsWorks Stacks intégrés en exécutant des livres de recettes personnalisés. Pour remplacer des attributs dans vos recettes ou dans celles de la communauté, suivez les instructions et les exemples de la section À propos des attributs dans la documentation de Chef 12.

- AWS OpsWorks Stacks continue de prendre en charge les couches suivantes pour les stacks Linux Chef 12 :
  - Couches personnalisées
  - Couche de service Amazon RDS
  - Couches de cluster ECS
- La configuration de la pile et les conteneurs de données pour Chef 12 Linux ont changé et ressemblent désormais beaucoup à leurs homologues de Chef 12.2 Windows. Cela facilite les requêtes, les analyses et la résolutions des problèmes de ces conteneurs de données, surtout si vous travaillez avec des piles utilisant des types de système d'exploitation différents. Notez que AWS OpsWorks Stacks ne prend pas en charge les sacs de données cryptés. Pour stocker des données sensibles sous une forme chiffrée, telle que mots de passe ou certificats, nous vous recommandons de les stocker dans un compartiment S3 privé. Vous pouvez ensuite créer une recette personnalisée qui utilise le Kit SDK Amazon pour Ruby pour récupérer les données. Pour obtenir un exemple, consultez Utilisation du kit SDK pour Ruby. Pour plus d'informations, consultez AWS OpsWorks Référence du sac de données Stacks.
- Dans Chef 12 Linux, Berkshelf n'est plus installé sur les instances de la pile. En revanche, nous vous recommandons d'utiliser Berkshelf sur un ordinateur de développement local créer le package de vos dépendances de livres de recettes localement. Téléchargez ensuite votre package, avec les dépendances incluses, sur Amazon Simple Storage Service. Enfin, modifiez votre pile Chef 12 Linux pour utiliser le package chargé comme source de livre de recettes. Pour de plus amples informations, veuillez consulter Empaquetage local des dépendances des livres de recettes.
- Les configurations RAID des volumes EBS ne sont plus prises en charge. Pour améliorer les performances, vous pouvez utiliser des IOPS provisionnées pour Amazon Elastic Block Store (Amazon EBS).
- autofs n'est plus pris en charge.
- Les référentiels Subversion ne sont plus pris en charge.
- Les installations des packages de système d'exploitation par couche doivent maintenant être effectuées avec des recettes personnalisées. Pour de plus amples informations, veuillez consulter Installations de package par couche.

Systèmes d'exploitation pris en charge

Chef 12 prend en charge les mêmes systèmes d'exploitation Linux que les versions antérieures de Chef. Pour obtenir la liste des types de système d'exploitation Linux et des versions susceptibles d'être utilisées par les piles Chef 12 Linux, consultez Systèmes d'exploitation Linux.

Types d'instance pris en charge

AWS OpsWorks Stacks prend en charge tous les types d'instances pour les piles Chef 12 Linux, à l'exception des types d'instances spécialisés tels que le calcul en cluster HPC, le GPU en cluster et les types d'instances de cluster à mémoire élevée.

En savoir plus

Pour en savoir plus sur la façon de travailler avec Chef 12 pour les piles Linux, consultez les ressources suivantes :

- Mise en route : exemple

  Vous présente AWS OpsWorks Stacks en vous guidant à travers un bref exercice pratique avec la console AWS OpsWorks Stacks pour créer un environnement d'application Node.js.

- Mise en route : Linux

  Vous présente AWS OpsWorks Stacks et Chef 12 Linux en vous guidant à travers un exercice pratique avec la console AWS OpsWorks Stacks pour créer une pile Linux Chef 12 de base contenant une couche simple avec une application Node.js qui gère le trafic.

- Couches personnalisées

  Donne des conseils pour ajouter une couche qui contient des livres de recettes et les recettes à une pile Linux Chef 12. Vous pouvez utiliser les livres de recettes et les recettes disponibles fournis par la communauté de Chef ou vous pouvez créer les vôtres.

- Passage aux conteneurs de données

  Compare et fait la différence entre le JSON d'instance utilisé par des piles Linux exécutant Chef 11 et des versions antérieures au JSON utilisé par Chef 12. Pointe également vers la documentation de référence pour le format JSON de l'instance Chef 12.

Transfert des paramètres de pile des attributs aux conteneurs de données

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks propose une grande variété de paramètres de stack pour vos recettes Chef. Ces paramètres de la pile incluent différentes valeurs :

- Source de livre de recettes Stack URLs
- Configurations des volumes de couche
- Noms d'hôte d'instance
- Noms DNS d'Elastic Load Balancing
- Source de l'application URLs
- Noms utilisateur

La référence des paramètres de pile depuis les recettes rend le code plus robuste et moins enclin aux erreurs que le codage en dur des paramètres de pile directement dans les recettes. La rubrique décrit comment accéder à ces paramètres de pile et comment passer des attributs de Chef 11.10 et versions antérieures pour Linux aux conteneurs de données dans Chef 12 Linux.

Dans Chef 11.10 et versions antérieures pour Linux, les paramètres de pile sont accessibles en tant qu'attributs Chef et via l'objet `node` de Chef ou la recherche Chef. Ces attributs sont stockés sur les instances AWS OpsWorks Stacks dans un ensemble de fichiers JSON du `/var/lib/aws/opsworks/chef` répertoire. Pour de plus amples informations, veuillez consulter Attributs de déploiement et de configuration de pile : Linux.

Dans Chef 12 Linux, les paramètres de pile sont accessibles en tant que conteneurs de données Chef et sont accessibles uniquement par le biais de la recherche Chef. Les sacs de données sont stockés sur les instances de AWS OpsWorks Stacks dans un ensemble de fichiers JSON du `/var/chef/runs/`*`run-ID`*`/data_bags` répertoire, où se *`run-ID`* trouve un identifiant unique que AWS

OpsWorks Stacks attribue à chaque exécution de Chef sur une instance. Les paramètres de pile n'étant plus accessibles en tant qu'attributs Chef, les paramètres de pile ne sont plus accessibles via l'objet node de Chef. Pour plus d'informations, consultez le AWS OpsWorks Référence du sac de données Stacks.

Par exemple, dans Chef 11.10 et versions antérieures de Linux, le code suivant de la recette utilise l'objet node de Chef pour obtenir les attributs représentant le nom court et l'URL source d'une application. Il utilise ensuite le journal Chef pour écrire ces deux valeurs d'attribut :

```
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']}' **********")
```

Dans Chef 12 Linux, le code suivant de la recette utilise l'index de recherche `aws_opsworks_app` pour obtenir le contenu du premier élément du conteneur de données `aws_opsworks_app`. Le code écrit ensuite deux messages dans le message Chef, l'un avec le contenu du conteneur de données du nom court de l'application, et l'autre avec le contenu du conteneur de données de l'URL source de l'application :

```
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']}' **********")
```

Pour migrer votre code de recette qui accède aux paramètres de pile depuis Chef 11.10 et versions antérieures pour Linux jusqu'à Chef 12 Linux, vous devez modifier votre code pour :

- Accéder aux conteneurs de données Chef au lieu des attributs Chef.
- Utiliser la recherche Chef au lieu de l'objet node de Chef.
- Utilisez des noms de sacs de données AWS OpsWorks Stacks tels que`aws_opsworks_app`, au lieu d'utiliser des noms d'attributs AWS OpsWorks Stacks tels que `opsworks` et. `deploy`

Pour plus d'informations, consultez le AWS OpsWorks Référence du sac de données Stacks.

Support pour les versions précédentes de Chef dans AWS OpsWorks Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section fournit un bref aperçu de la documentation AWS OpsWorks Stacks pour les versions précédentes de Chef.

Chef 11.10 et versions antérieures pour Linux

Fournit de la documentation sur le support de AWS OpsWorks Stacks pour Chef 11.10, 11.4 et 0.9 pour les stacks Linux.

Chef 11.10 et versions antérieures pour Linux

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section fournit un bref aperçu de la documentation AWS OpsWorks Stacks pour Chef 11.10, 11.4 et 0.9 pour Linux.

Mise en route des piles Linux Chef 11

Fournit une procédure qui montre comment créer une pile de serveur d'applications PHP simple mais fonctionnelle.

[Création de votre première pile Node.js]

Décrit comment créer une pile Linux qui prend en charge un serveur d'applications Node.js et comment déployer une application simple.

[Personnalisation des piles AWS OpsWorks]

Décrit comment personnaliser AWS OpsWorks Stacks pour répondre à vos besoins spécifiques.

[Les bases des livres de recettes]

Décrit comment implémenter des recettes pour les instances AWS OpsWorks Stacks.

[Equilibrage de charge d'une couche]

Décrit comment utiliser les options d'équilibrage de charge disponibles dans AWS OpsWorks Stacks.

[Running a Stack in a VPC]

Décrit comment créer et exécuter une pile dans un Virtual Private Cloud.

[Migration depuis Chef Server]

Fournit des directives pour la migration de Chef Server vers AWS OpsWorks Stacks.

[AWS OpsWorks Référence de la couche Stacks]

Décrit les couches intégrées AWS OpsWorks Stacks disponibles.

[Composants des livres de recettes]

Décrit les trois composants standard des livres de recettes : attributs, modèles et recettes.

[Attributs de déploiement et de configuration de pile : Linux]

Décrit les attributs de déploiement et de configuration de pile pour Linux.

[Attributs des livres de recettes intégrés]

Explique comment utiliser les attributs de la recette intégrée afin de contrôler la configuration des logiciels installés.

[Résolution des problèmes de Chef 11.10 et versions antérieures pour Linux]

Décrit les approches permettant de résoudre divers problèmes dans AWS OpsWorks Stacks.

Mise en route des piles Linux Chef 11

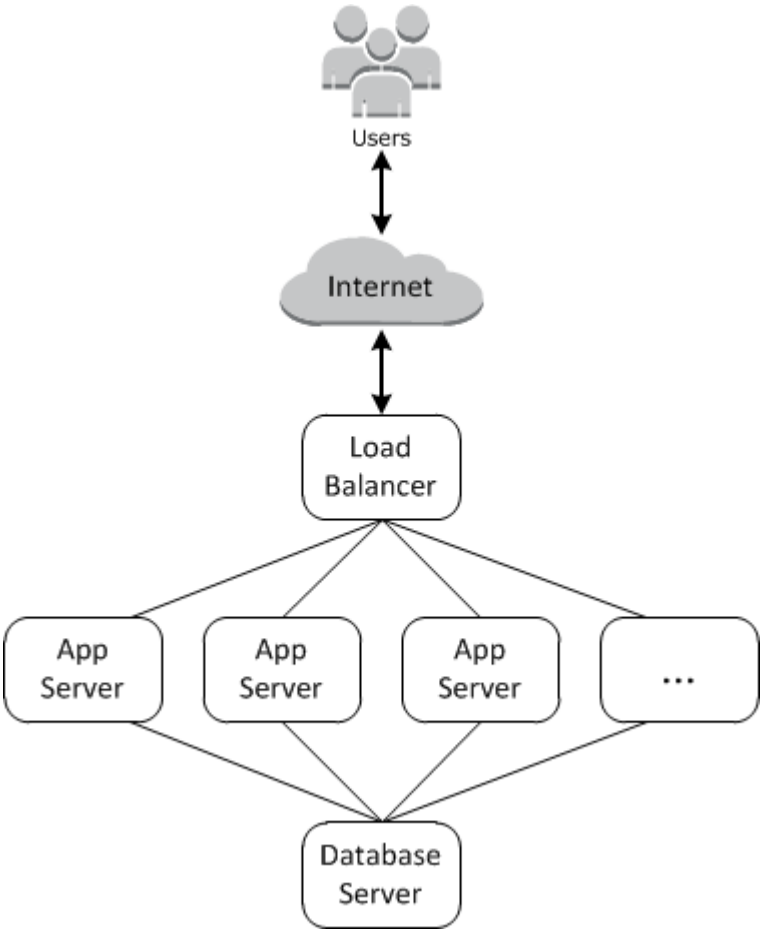
Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Note

Cette section décrit comment démarrer avec les piles Linux à l'aide de Chef 11. Pour plus d'informations sur la mise en route avec les piles Linux de Chef 12, consultez Mise en route : Linux. Pour plus d'informations sur la mise en route avec les piles Windows de Chef 12, consultez Mise en route : Windows.

Les applications basées sur le cloud nécessitent généralement un groupe de ressources connexes (serveurs d'applications, serveurs de base de données, etc.) qui doivent être créées et gérées collectivement. Cet ensemble d'instances est appelé une pile. Une simple pile d'application peut se présenter comme suit.



L'architecture de base comprend les éléments suivants :

- Un équilibreur de charge pour répartir équitablement le trafic entrant des utilisateurs sur les serveurs d'applications.
- Un ensemble d'instances serveur d'applications, en nombre suffisant pour gérer le trafic.
- Un serveur de base de données pour fournir aux serveurs d'applications un magasin de données principal.

En outre, vous avez généralement besoin d'un moyen de distribution des applications sur les serveurs d'applications, de surveillance de la pile, etc.

AWS OpsWorks Stacks fournit un moyen simple et direct de créer et de gérer des piles ainsi que les applications et ressources associées. Ce chapitre présente les principes de base de AWS OpsWorks Stacks, ainsi que certaines de ses fonctionnalités les plus sophistiquées, en vous expliquant le processus de création de la pile de serveurs d'applications dans le diagramme. Il utilise un modèle de développement incrémentiel que AWS OpsWorks Stacks rend facile à suivre : configurez une pile

de base et, une fois qu'elle fonctionne correctement, ajoutez des composants jusqu'à obtenir une implémentation complète.

- Étape 1 : Exécuter les opérations prérequises montre comment se préparer à démarrer la procédure pas à pas.
- Étape 2 : Créer une pile de serveur d'applications simple - Chef 11 montre comment créer une pile minimale qui se compose d'un seul serveur d'application.
- Étape 3 : Ajouter un magasin de données principal montre comment ajouter un serveur de base de données et le connecter au serveur d'applications.
- Étape 4 : Diminution MyStack montre comment dimensionner une pile de façon à gérer la charge accrue en ajoutant plusieurs serveurs d'applications et un équilibreur de charge afin de répartir le trafic entrant.

Rubriques

- Étape 1 : Exécuter les opérations prérequises
- Étape 2 : Créer une pile de serveur d'applications simple - Chef 11
- Étape 3 : Ajouter un magasin de données principal
- Étape 4 : Diminution MyStack
- Étape 5 : Supprimer MyStack

Étape 1 : Exécuter les opérations prérequises

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Complétez les étapes de configuration suivantes avant de commencer la procédure pas à pas. Ces étapes de configuration incluent la création d'un AWS compte, la création d'un utilisateur administratif et l'attribution d'autorisations d'accès à AWS OpsWorks Stacks.

Si vous avez déjà terminé l'une des procédures Débuter avec AWS OpsWorks Stacks, vous avez rempli les conditions requises pour cette procédure et vous pouvez passer directement à Étape 2 : Créer une pile de serveur d'applications simple - Chef 11.

Rubriques

- Inscrivez-vous pour un Compte AWS
- Création d'un utilisateur doté d'un accès administratif
- Attribuez des autorisations d'accès au service à votre utilisateur

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

   Une partie de la procédure d'inscription consiste à recevoir un appel téléphonique ou un message texte et à saisir un code de vérification sur le clavier du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à https://aws.amazon.com/et en choisissant Mon compte.

Création d'un utilisateur doté d'un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez Utilisateur racine d'un compte AWS AWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez Connexion en tant qu'utilisateur racine dans le Guide de l'utilisateur Connexion à AWS .
2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, consultez la section Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d' AWS IAM Identity Center dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu'utilisateur doté d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

  Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribution d'un accès à d'autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d'autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

Pour obtenir des instructions, consultez Création d'un ensemble d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

   Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l'utilisateur AWS IAM Identity Center .

Attribuez des autorisations d'accès au service à votre utilisateur

Activez l'accès au service AWS OpsWorks Stacks (et aux services connexes sur lesquels AWS OpsWorks Stacks s'appuie) en ajoutant les `AmazonS3FullAccess` autorisations `AWSOpsWorks_FullAccess` et à votre rôle ou à votre utilisateur.

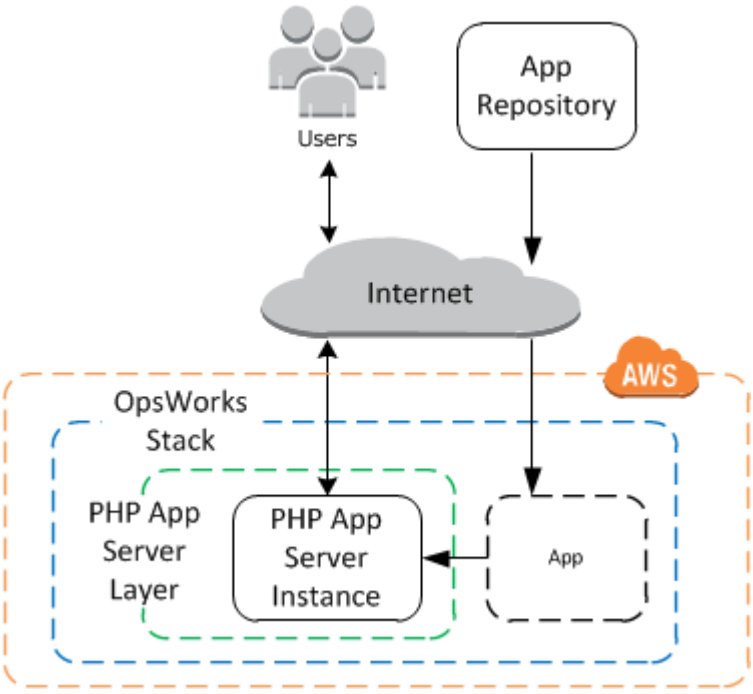
Pour plus d'informations sur l'ajout d'autorisations, consultez la section Ajout d'autorisations d'identité IAM (console).

Vous avez maintenant terminé toutes les étapes de configuration et pouvez démarrer cette procédure pas à pas.

Étape 2 : Créer une pile de serveur d'applications simple - Chef 11

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une pile serveur d'applications de base se compose d'une seule instance serveur d'applications avec une adresse IP publique pour recevoir les demandes utilisateur. Le code de l'application et les fichiers associés sont stockés dans un référentiel distinct et déployés à partir de là vers le serveur. Le schéma suivant illustre une telle pile.



La pile comprend les éléments suivants :

- Une couche, qui représente un groupe d'instances et spécifie comment elles doivent être configurées.

  Dans cet exemple, la couche représente un groupe d'instances de PHP App Server.
- Une instance, qui représente une EC2 instance Amazon.

  Dans ce cas, l'instance est configurée pour exécuter un serveur d'applications PHP. Les couches peuvent avoir autant d'instances que vous le souhaitez. AWS OpsWorks Stacks prend également en charge plusieurs autres serveurs d'applications. Pour de plus amples informations, veuillez consulter Couches serveur d'applications.
- Une application qui contient les informations requises pour installer une application sur le serveur d'applications.

  Le code est stocké dans un référentiel distant, tel qu'un référentiel Git ou un compartiment Amazon S3.

Les sections suivantes décrivent comment utiliser la console AWS OpsWorks Stacks pour créer la pile et déployer l'application. Vous pouvez également utiliser un AWS CloudFormation modèle pour approvisionner une pile. Pour un exemple de modèle fournissant la pile décrite dans cette rubrique, consultez AWS OpsWorks Snippets.

Rubriques

- Étape 2.1 : Créer une pile - Chef 11
- Étape 2.2 : Ajouter une couche de serveur d'applications PHP - Chef 11
- Étape 2.3 : Ajouter une instance à la couche PHP App Server - Chef 11
- Étape 2.4 : Créer et déployer une application - Chef 11

Étape 2.1 : Créer une pile - Chef 11

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous démarrez un projet AWS OpsWorks Stacks en créant une pile, qui fait office de conteneur pour vos instances et autres ressources. La configuration de la pile spécifie certains paramètres de base, telles que la région AWS et le système d'exploitation par défaut, qui sont partagés par toutes les instances de la pile.

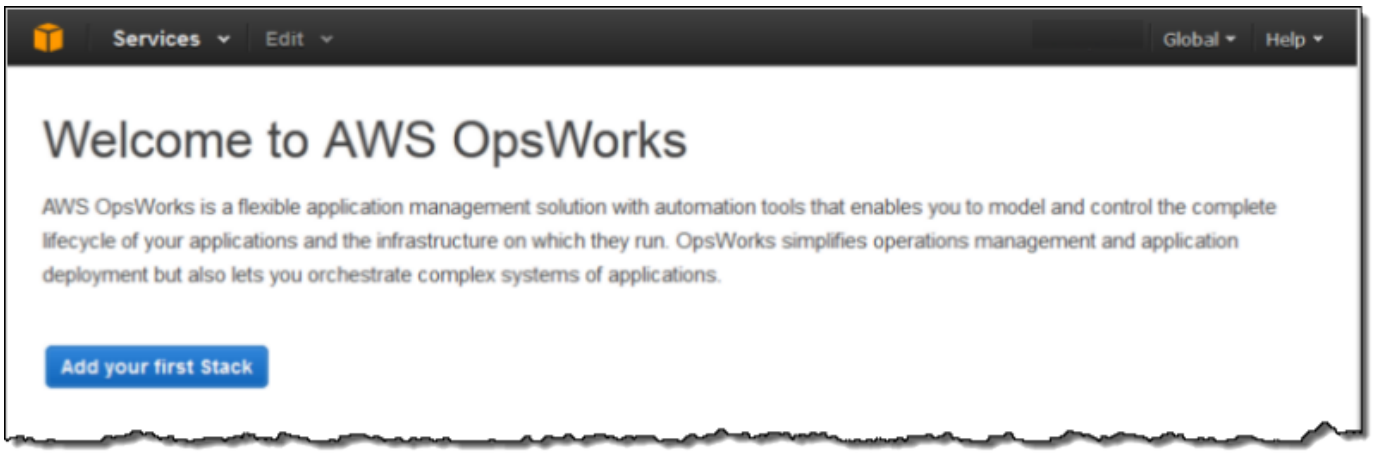
> **Note**
>
> Cette page vous permet de créer des piles Chef 11. Pour plus d'informations sur la création des piles Chef 12, consultez Créer une pile.

Cette page vous permet de créer des piles dans Chef 11.

Pour créer une pile

1. Ajouter une pile

   Connectez-vous à la console AWS OpsWorks Stacks. Si le compte ne possède aucune pile existante, vous verrez la OpsWorks page Bienvenue sur AWS ; cliquez sur Ajouter votre première pile. Sinon, vous verrez le tableau de bord AWS OpsWorks Stacks, qui répertorie les piles de votre compte ; cliquez sur Ajouter une pile.



2. Configurer la pile

   Sur la page Add Stack (Ajouter une pile), choisissez Chef 11 stack (Pile Chef 11) et spécifiez les paramètres suivants :

   Nom de la pile

   Entrez un nom pour votre pile, qui peut contenir des caractères alphanumériques (a—z, A—Z et 0—9) et des tirets (-). L'exemple de pile de cette procédure est nommé **`MyStack`**.

   Région

   Sélectionnez USA West (Oregon) comme région de la pile.

   Acceptez les valeurs par défaut pour les autres paramètres, puis cliquez sur Add Stack (Ajouter une pile). Pour plus d'informations sur les différents paramètres de pile, consultez Créer une pile.

Étape 2.2 : Ajouter une couche de serveur d'applications PHP - Chef 11

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Même si une pile est essentiellement un conteneur d'instances, vous n'ajoutez pas directement les instances à une pile. Vous ajoutez une couche qui représente un groupe d'instances connexes, puis vous ajoutez des instances à la couche.

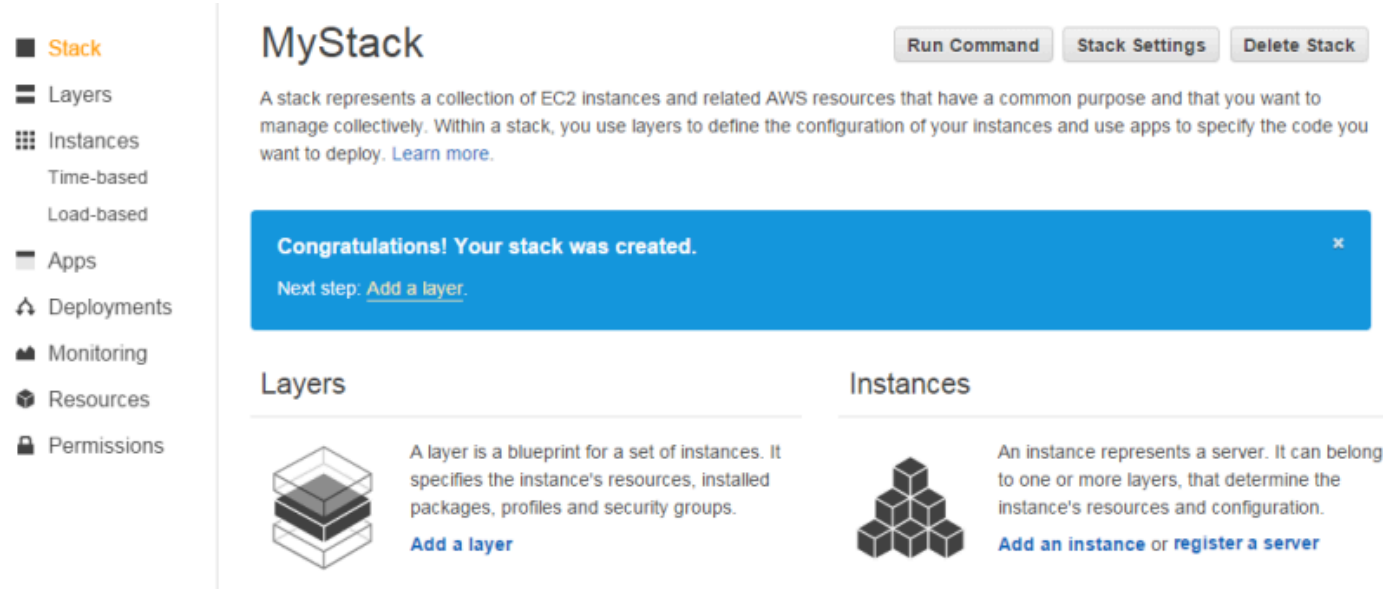
Une couche est essentiellement un plan que AWS OpsWorks Stacks utilise pour créer un ensemble d' EC2 instances Amazon avec la même configuration. Vous ajoutez une couche à la pile pour chaque groupe d'instances associées. AWS OpsWorks Stacks inclut un ensemble de couches intégrées pour représenter des groupes d'instances exécutant des progiciels standard tels qu'un serveur de base de données MySQL ou un serveur d'applications PHP. En outre, vous pouvez créer des couches totalement ou partiellement personnalisées en fonction de vos besoins. Pour de plus amples informations, veuillez consulter Personnalisation des piles AWS OpsWorks.

MyStack possède une couche, la couche PHP App Server intégrée, qui représente un groupe d'instances qui fonctionnent comme des serveurs d'applications PHP. Pour plus d'informations, y compris les descriptions des couches intégrées, consultez Couches.

Pour ajouter une couche PHP App Server à MyStack

1. Ouvrir la page Add Layer

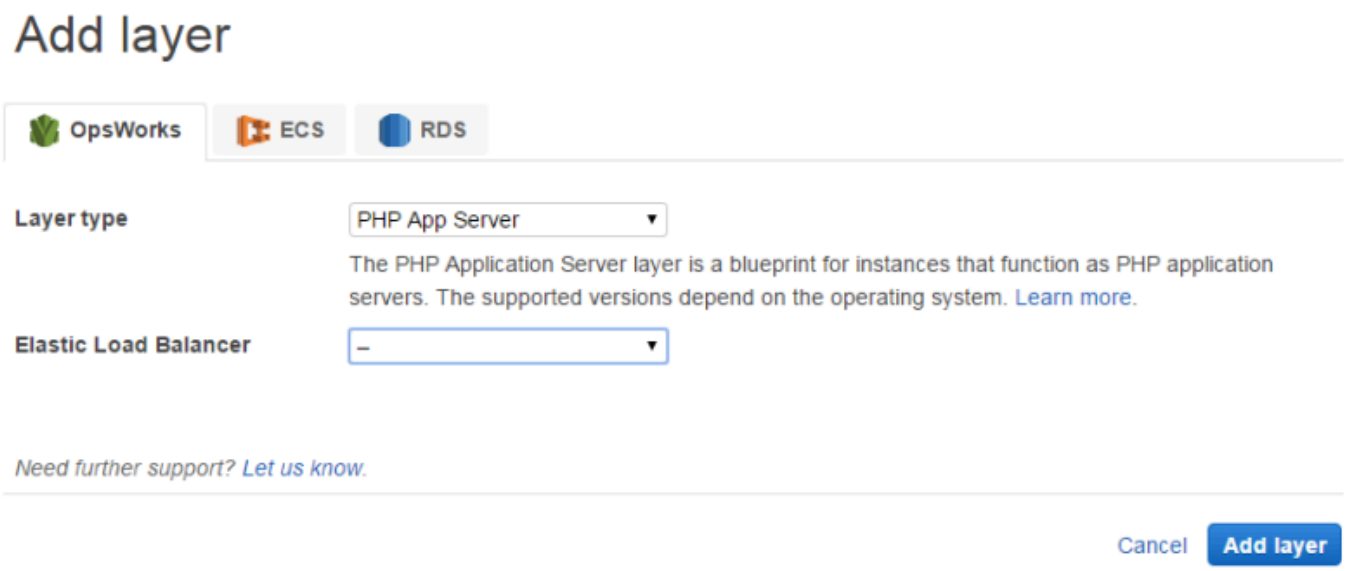
   Une fois que vous avez fini de créer la pile, AWS OpsWorks Stacks affiche la page Stack. Cliquez sur Add a layer (Ajouter une couche) pour ajouter votre première couche.



2. Spécifier un type de couche et configurer la couche

   Dans le champ Type de couche, sélectionnez PHP App Server, acceptez le paramètre par défaut d'Elastic Load Balancer et cliquez sur Ajouter une couche. Une fois que vous avez créé

la couche, vous pouvez spécifier d'autres attributs tels que la configuration du volume EBS en modifiant la couche.



Étape 2.3 : Ajouter une instance à la couche PHP App Server - Chef 11

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une instance AWS OpsWorks Stacks représente une EC2 instance Amazon particulière :

- La configuration de l'instance spécifie quelques éléments de base tels que le EC2operating système Amazon et sa taille ; elle fonctionne mais ne fait pas grand-chose.
- La couche de l'instance ajoute des fonctionnalités à l'instance en déterminant quels packages doivent être installés, si l'instance a une adresse IP Elastic, etc.

AWS OpsWorks Stacks installe un agent sur chaque instance qui interagit avec le service. Pour ajouter les fonctionnalités d'une couche à une instance, AWS OpsWorks Stacks demande à l'agent d'exécuter de petites applications appelées recettes Chef, qui peuvent installer des applications et

des packages, créer des fichiers de configuration, etc. AWS OpsWorks Stacks exécute des recettes à des moments clés du cycle de vie de l'instance. Par exemple, OpsWorks exécute les recettes d'installation une fois le démarrage de l'instance terminé pour gérer des tâches telles que l'installation du logiciel, et exécute les recettes de déploiement lorsque vous déployez une application pour installer le code et les fichiers associés.

> **Note**
>
> Si vous êtes curieux de savoir comment fonctionnent les recettes, toutes les recettes intégrées à AWS OpsWorks Stacks se trouvent dans un GitHub référentiel public : OpsWorks Cookbooks. Vous pouvez également créer vos propres recettes personnalisées et demander à AWS OpsWorks Stacks de les exécuter, comme décrit plus tard.

Pour ajouter un serveur d'applications PHP MyStack, ajoutez une instance à la couche PHP App Server que vous avez créée à l'étape précédente.

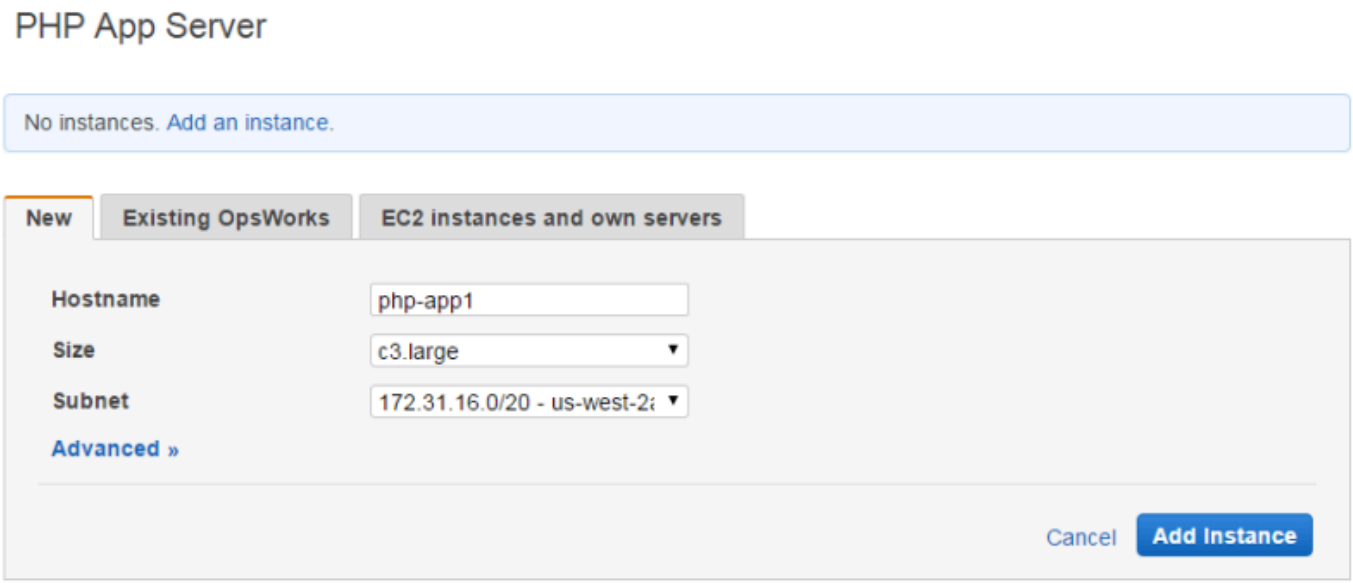
Pour ajouter une instance à la couche PHP App Server

1. Ouvrir Add an Instance

   Une fois que vous avez fini d'ajouter la couche, AWS OpsWorks Stacks affiche la page Couches. Cliquez sur Instances dans le volet de navigation, puis sous PHP App Server, cliquez sur Ajouter une instance.

2. Configurer l'instance

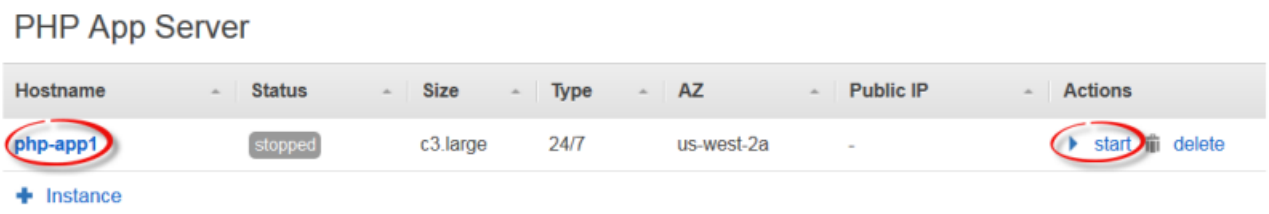
   Chaque instance possède un nom d'hôte par défaut qui est généré pour vous par AWS OpsWorks Stacks. Dans cet exemple, AWS OpsWorks Stacks ajoute simplement un numéro au nom abrégé de la couche. Vous pouvez configurer chaque instance séparément, y compris le remplacement de certains paramètres par défaut que vous avez spécifiés lors de la création de la pile, par exemple le système d'exploitation ou la zone de disponibilité. Pour cette procédure, acceptez simplement les paramètres par défaut et cliquez sur Add Instance (Ajouter une instance) pour ajouter l'instance à la couche. Pour de plus amples informations, veuillez consulter instances.



3. Démarrer l'instance

   Jusqu'ici, vous avez spécifié la configuration de l'instance. Vous devez démarrer une instance pour créer une EC2 instance Amazon en cours d'exécution. AWS OpsWorks Stacks utilise ensuite les paramètres de configuration pour lancer une EC2 instance Amazon dans la zone de disponibilité spécifiée. Les détails du lancement d'une instance dépendent du type de dimensionnement de l'instance. Au cours de l'étape précédente, vous avez créé une instance avec le type de dimensionnement par défaut 24/7, qui doit être démarrée manuellement, puis est exécutée jusqu'à son arrêt manuel. Vous pouvez également créer des types de dimensionnement basés sur le temps et sur la charge, que AWS OpsWorks Stacks démarre et arrête automatiquement en fonction d'un calendrier ou de la charge actuelle. Pour de plus amples informations, veuillez consulter Gestion de la charge avec des instances basées sur le temps et sur la charge.

   Accédez à php-app1 sous PHP App Server et cliquez sur Démarrer dans la colonne Actions de la ligne pour démarrer l'instance.

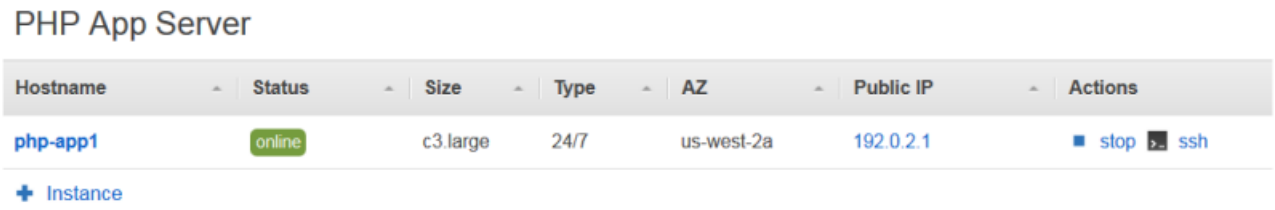


4. Surveiller l'état de l'Instance au moment du démarrage

   Le démarrage de l' EC2 instance Amazon et l'installation des packages prennent généralement quelques minutes. Au fur et à mesure que le démarrage progresse, le champ Status (Statut) de l'instance affiche la série de valeurs suivantes :

   1. demandé - AWS OpsWorks Stacks a appelé le EC2 service Amazon pour créer l' EC2 instance Amazon.
   2. en attente : AWS OpsWorks Stacks attend le démarrage de l' EC2instance Amazon.
   3. démarrage : l' EC2 instance Amazon est en train de démarrer.
   4. running_setup - L'agent AWS OpsWorks Stacks exécute les recettes de configuration de la couche, qui gèrent des tâches telles que la configuration et l'installation de packages, et les recettes de déploiement, qui déploient toutes les applications sur l'instance.
   5. online (en ligne) - l'instance est prête à être utilisée.

   Une fois que php-app1 est en ligne, la page Instances doit ressembler à ce qui suit :



   La page commence par un bref résumé de toutes les instances de votre pile. Maintenant, elle montre une instance en ligne. Dans la colonne Actions de php-app1, notez que l'option stop, qui arrête l'instance, a remplacé start et delete.

Étape 2.4 : Créer et déployer une application - Chef 11

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Pour être MyStack plus utile, vous devez déployer une application sur l'instance PHP App Server. Vous stockez le code d'une application et tous les fichiers associés dans un référentiel, par exemple Git. Vous devez suivre quelques étapes pour faire arriver ces fichiers sur vos serveurs d'applications :

> **Note**
>
> La procédure de cette section s'applique aux piles Chef 11. Pour obtenir des informations sur l'ajout d'applications aux couches des piles Chef 12, consultez Ajout d'applications.

1. Crée une application.

   Une application contient les informations dont AWS OpsWorks Stacks a besoin pour télécharger le code et les fichiers associés depuis le référentiel. Vous pouvez également spécifier des informations supplémentaires, telles que le domaine de l'application.
2. Déployez l'application sur vos serveurs d'applications.

   Lorsque vous déployez une application, AWS OpsWorks Stacks déclenche un événement du cycle de vie de déploiement. L'agent exécute ensuite les recettes Deploy de l'instance, qui téléchargent les fichiers dans le répertoire approprié avec les tâches connexes, telles que la configuration du serveur, le redémarrage du service, etc.

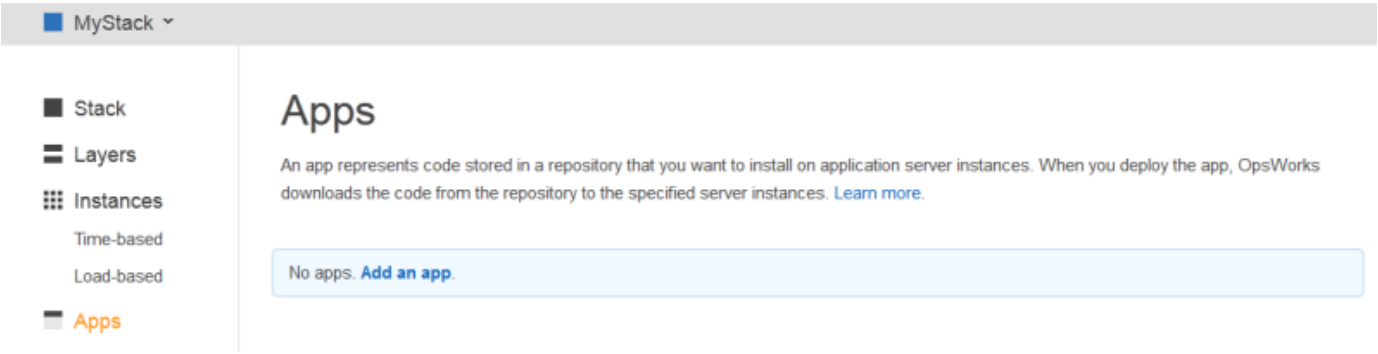
> **Note**
>
> Lorsque vous créez une nouvelle instance, AWS OpsWorks Stacks déploie automatiquement toutes les applications existantes sur l'instance. Toutefois, lorsque vous créez une application ou que vous mettez à jour une application existante, vous devez déployer manuellement l'application ou mettre à jour toutes les instances existantes.

Cette étape montre comment déployer manuellement un exemple d'application à partir d'un référentiel Git public vers un serveur d'applications. Si vous souhaitez examiner l'application, accédez à https://github.com/amazonwebservices/opsworks-demo-php-simple-app. L'application utilisée dans cet exemple se trouve dans la branche version1. AWS OpsWorks Stacks prend également en charge plusieurs autres types de référentiels. Pour de plus amples informations, veuillez consulter Source de l'application.

Pour créer et déployer une application

1. Ouvrir la page Apps

   Dans le panneau de navigation, cliquez sur Apps (Applications) et sur la page Apps (Applications), cliquez sur Add an app (Ajouter une application).



2. Configurer l'application

   Sur la page App (Application), spécifiez les valeurs suivantes :

   Nom

   Le nom de l'application, que AWS OpsWorks Stacks utilise à des fins d'affichage. L'exemple d'application est nommé**`SimplePHPApp`**. AWS OpsWorks Stacks génère également un nom abrégé, simplephpapp pour cet exemple, qui est utilisé en interne et par les recettes de déploiement, comme décrit plus loin.

   Type

   Le type de l'application qui détermine où déployer l'application. L'exemple utilise PHP, qui déploie l'application sur les instances de PHP App Server.

   Type de source de données

   Un serveur de base de données associé. Pour l'instant, sélectionnez None (Aucun) ; nous présenterons les serveurs de base de données dans Étape 3 : Ajouter un magasin de données principal.

   Type de référentiel

   Le type de référentiel de l'application. L'exemple d'application est stocké dans un référentiel Git.

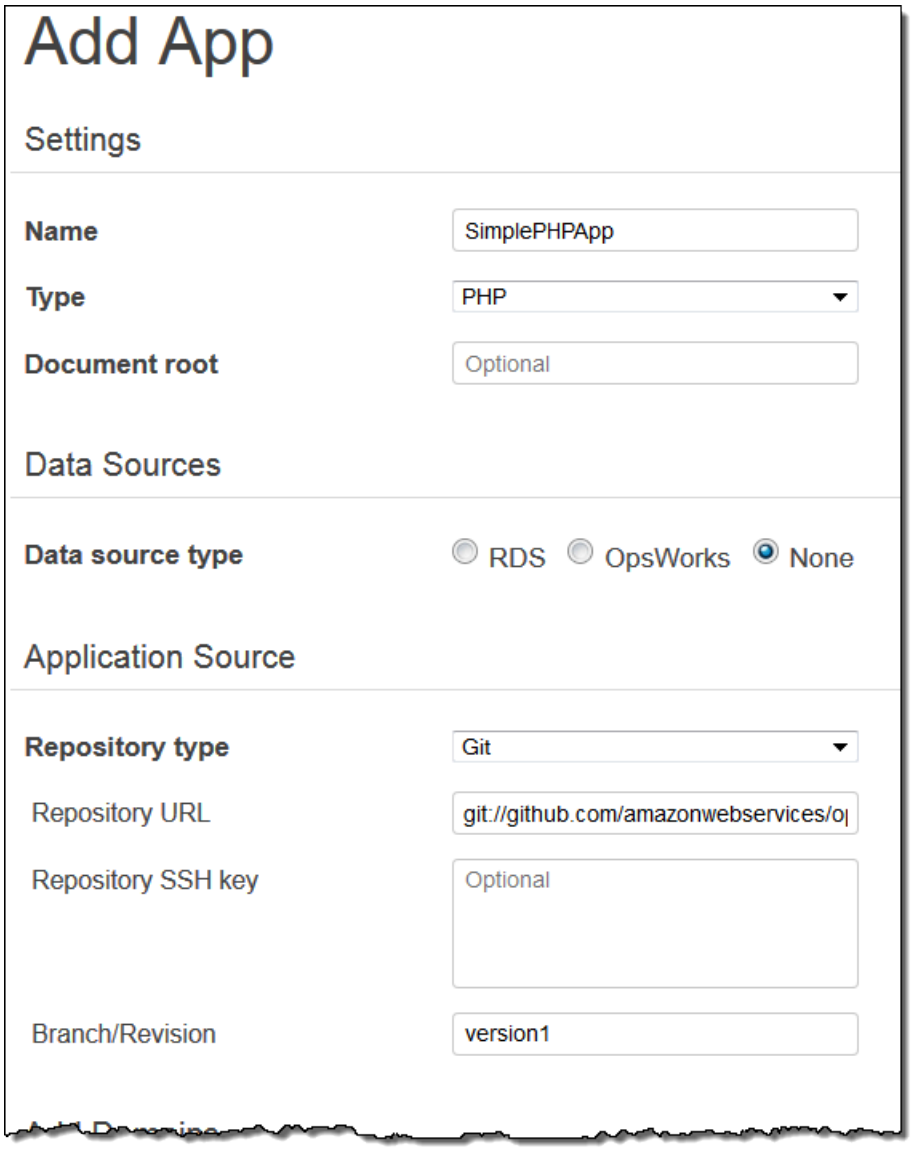
URL du référentiel

L'URL du référentiel de l'application. L'exemple d'URL est : **`git://github.com/awslabs/opsworks-demo-php-simple-app.git`**

Branch/Revision (Branche/Révision)

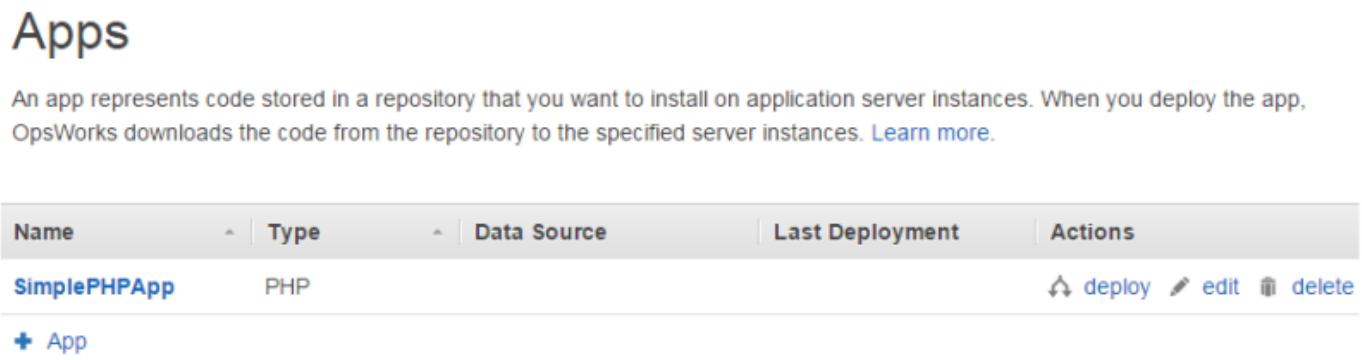
La branche ou la version de l'application. Cette partie de la procédure utilise la branche **`version1`**.

Conservez les valeurs par défaut pour les autres paramètres et cliquez sur Add App (Ajouter une application). Pour de plus amples informations, veuillez consulter Ajout d'applications.



3. Ouvrir la page Deployment

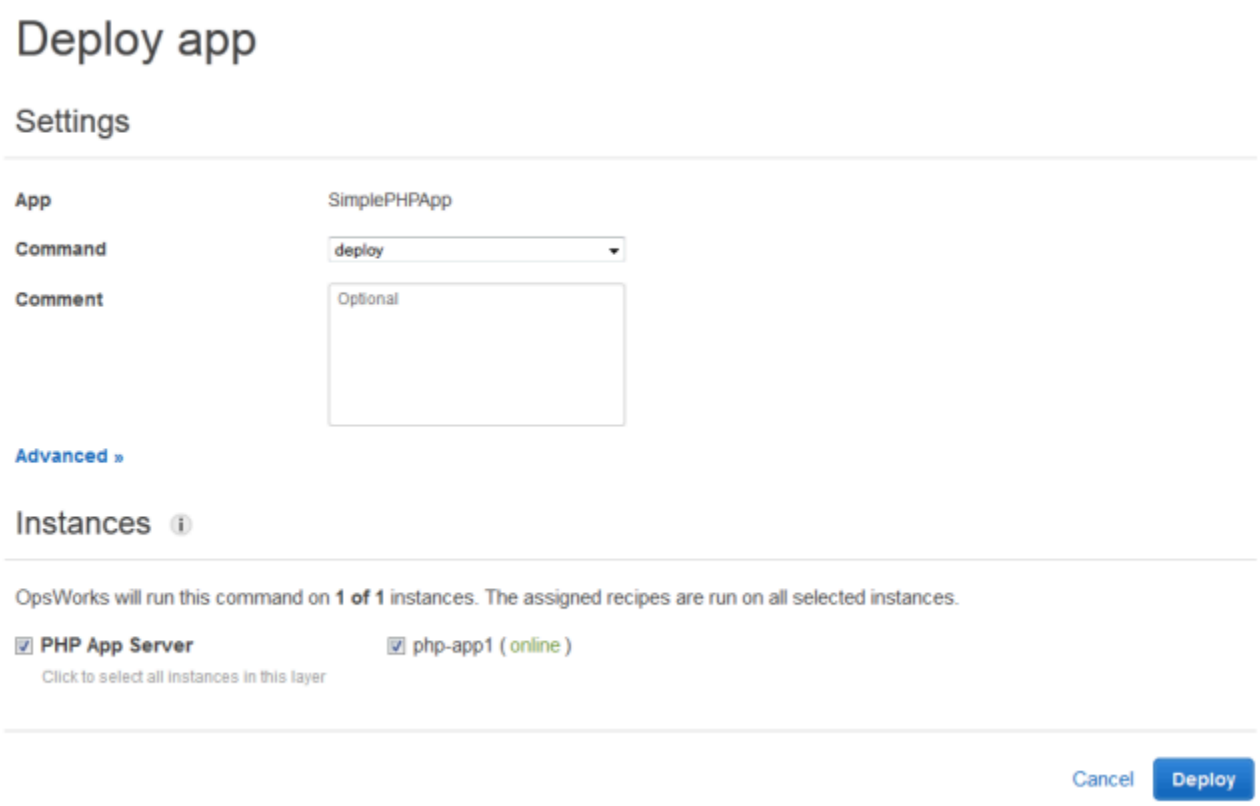
   Pour installer le code sur le serveur, vous devez déployer l'application. Pour ce faire, cliquez sur Déployer dans la colonne PHPApp Actions simples.



4. Déployer l'application

   Lorsque vous déployez une application, l'agent exécute les recettes de déploiement sur l'instance du serveur d'applications PHP, qui téléchargent et configurent l'application.

   Command (Commande) doit déjà être défini sur deploy. Conserver les valeurs par défaut pour les autres paramètres et cliquez sur Deploy (Déployer) pour déployer l'application.



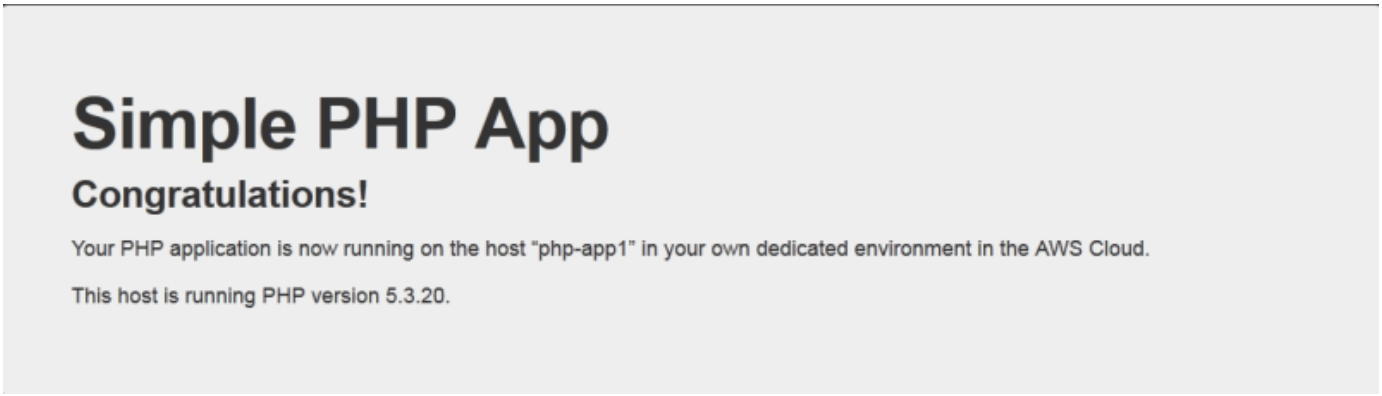
Une fois le déploiement terminé, la page Deployment (Déploiement) affiche un Status (Statut) Successful (Succès) et php-app1 est marquée d'une coche verte.

5. Exécutez simplement PHPApp

   Simple PHPApp est maintenant installé et prêt à fonctionner. Pour l'exécuter, cliquez sur Instances dans le panneau de navigation afin d'accéder à la page Instances. Cliquez ensuite sur l'adresse IP publique de l'instance php-app1.



   Vous devriez voir une page similaire à la suivante dans votre navigateur.

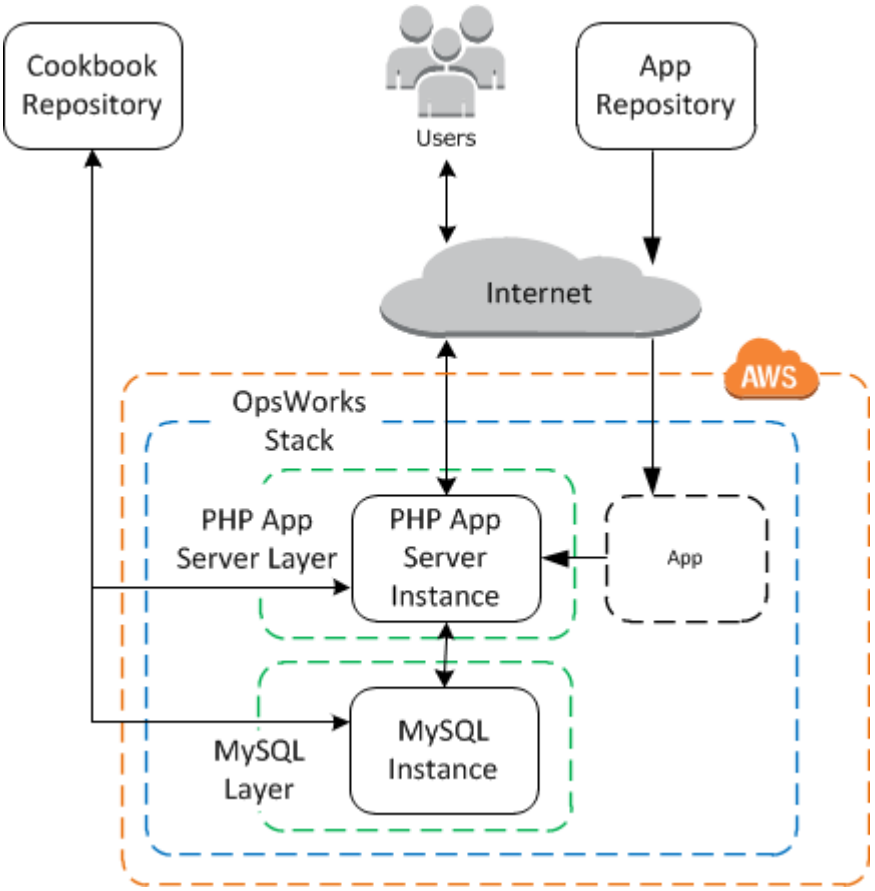


> **Note**
>
> Cette procédure suppose que vous passerez à la section suivante, puis que vous exécuterez toute la procédure au cours d'une seule session. Si vous préférez, vous pouvez vous arrêter à tout moment et continuer plus tard en vous connectant à AWS OpsWorks Stacks et en ouvrant le stack. Cependant, vous êtes facturé pour toutes les ressources AWS que vous utilisez, telles que des instances en ligne. Pour éviter des frais inutiles, vous pouvez arrêter votre instance, ce qui met fin à l' EC2instance correspondante. Vous pouvez démarrer les instances à nouveau lorsque vous êtes prêt à continuer.

Étape 3 : Ajouter un magasin de données principal

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Étape 2.1 : Créer une pile - Chef 11 vous a montré comment créer une pile ayant servi une application PHP. Cependant, il s'agissait d'une application très simple qui ne faisait guère plus qu'afficher un texte statique. Les applications en production utilisent généralement un magasin de données principal, ce qui donne une configuration de pile similaire à l'illustration suivante.



Cette section explique comment étendre MyStack pour inclure un serveur de base de données MySQL principal. Vous ne pouvez pas, toutefois, vous contenter d'ajouter un serveur MySQL à la pile. Vous devez également configurer l'application pour qu'elle communique correctement avec le serveur de base de données. AWS OpsWorks Stacks ne le fait pas pour vous ; vous devrez implémenter des recettes personnalisées pour effectuer cette tâche.

Rubriques

- Étape 3.1 : Ajouter une base de données principale
- Étape 3.2 : Mise à jour simple PHPApp
- Une courte digression : attributs des livres de cuisine, des recettes et des piles AWS OpsWorks
- Étape 3.3 : Ajoutez les livres de recettes personnalisés à MyStack
- Étape 3.4 : Exécuter les recettes
- Étape 3.5 : Déploiement de SimplePHPApp, version 2
- Étape 3.6 : Exécuter en toute simplicité PHPApp

Étape 3.1 : Ajouter une base de données principale

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

La nouvelle version de Simple PHPApp stocke ses données dans une base de données principale. AWS OpsWorks Stacks prend en charge deux types de serveurs de base de données :

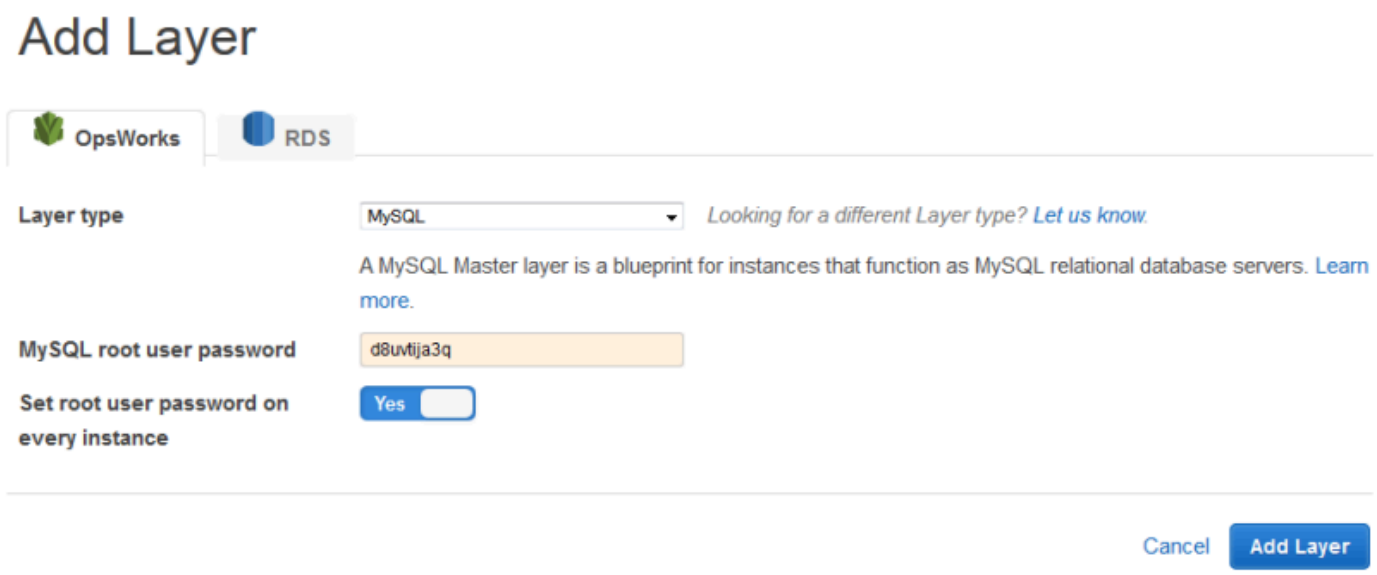
- La couche MySQL AWS OpsWorks Stacks est un modèle pour créer des EC2 instances Amazon hébergeant un maître de base de données MySQL.
- La couche de service Amazon RDS permet d'intégrer une instance Amazon RDS dans une pile.

Vous pouvez également utiliser d'autres bases de données, telles qu'Amazon DynamoDB, ou créer une couche personnalisée pour prendre en charge des bases de données telles que MongoDB. Pour de plus amples informations, veuillez consulter the section called “Utilisation d'un magasin de données principal”.

Cet exemple utilise une couche MySQL.

Pour ajouter une couche MySQL à MyStack

1. Sur la page Layers (Couches), cliquez sur + Layer (+ Couche).
2. Sur la page Add Layer (Ajouter une couche), pour Layer type (Type de couche), sélectionnez MySQL, acceptez les paramètres par défaut et cliquez sur Add Layer (Ajouter une couche).



Pour ajouter une instance à la couche MySQL

1. Sur la ligne MySQL de la page Layers, cliquez sur Ajouter une instance.
2. Sur la page Instances, sous MySQL, cliquez sur Add an instance (Ajouter une instance).
3. Acceptez les valeurs par défaut et cliquez sur Add instance (Ajouter une instance), mais ne démarrez pas encore.

> **Note**
>
> AWS OpsWorks Stacks crée automatiquement une base de données nommée en utilisant le nom abrégé de l'application, simplephpapp pour cet exemple. Vous aurez besoin de ce nom si vous souhaitez utiliser les recettes Chef pour interagir avec la base de données.

Étape 3.2 : Mise à jour simple PHPApp

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Pour commencer, vous avez besoin d'une nouvelle version de Simple PHPApp qui utilise un magasin de données principal. Avec AWS OpsWorks Stacks, il est facile de mettre à jour une application. Si vous utilisez un référentiel Git ou Subversion, vous pouvez avoir une branche de référentiel distincte pour chaque version de l'application. L'exemple stocke une version de l'application qui utilise une base de données principale de la branche version2 du référentiel Git. Il suffit de mettre à jour la configuration de l'application pour spécifier la nouvelle branche et redéployer l'application.

Pour mettre à jour Simple PHPApp

1. Ouvrir la page Edit de l'application

   Dans le volet de navigation, cliquez sur Applications, puis sur Modifier dans la colonne Actions de la PHPApp ligne Simple.
2. Mettre à jour la configuration de l'application

   Modifiez les paramètres suivants.

   Branch/Revision

   Ce paramètre indique la branche de référentiel de l'application. La première version de Simple PHPApp ne se connectait pas à une base de données. Pour utiliser une version base de données de l'application, définissez la valeur sur **`version2`**.

   Document root (Racine du document)

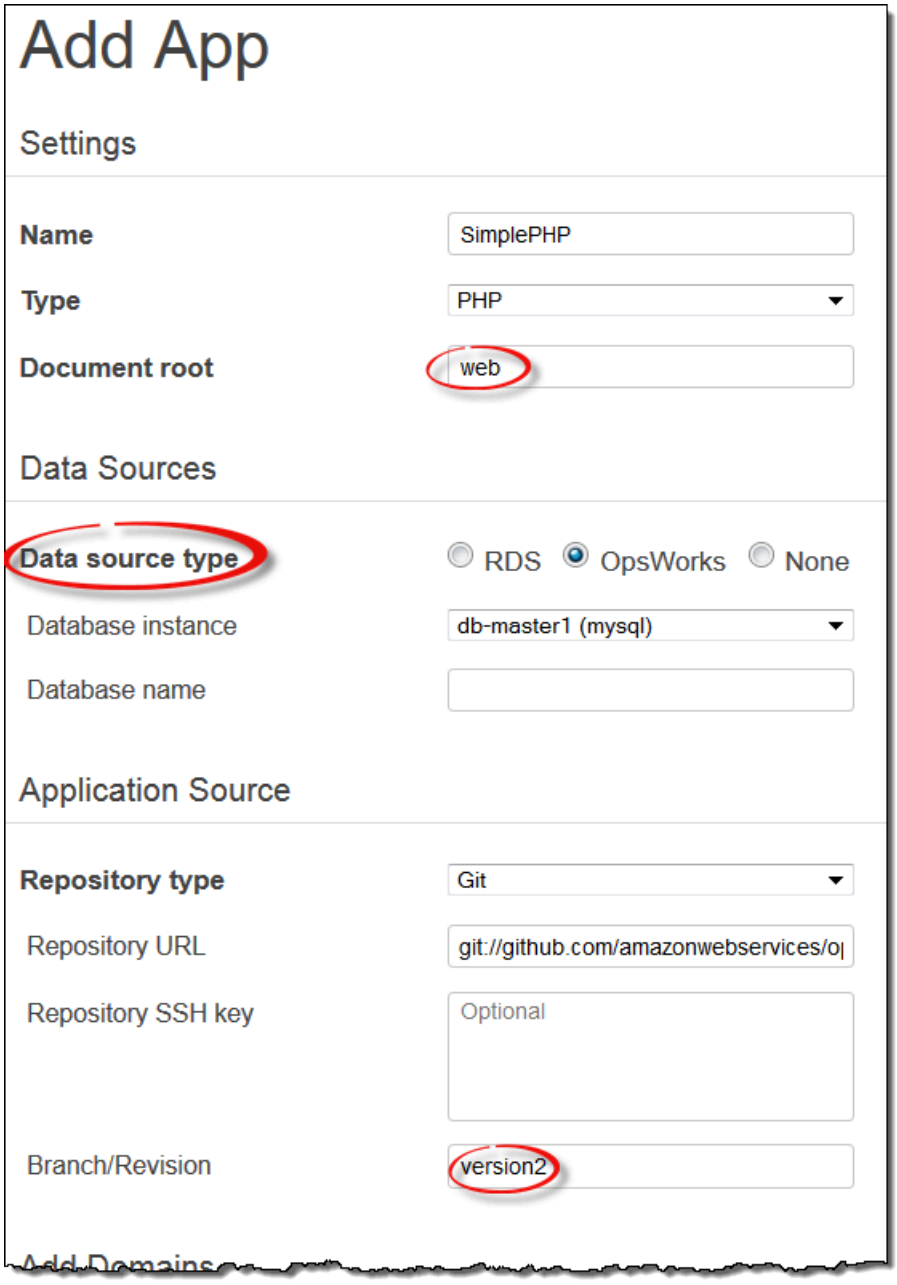
   Ce paramètre spécifie le dossier racine de votre application. La première version de Simple PHPApp utilisait le paramètre par défaut, qui s'installe `index.php` dans le dossier racine standard du serveur (`/srv/www`pour les applications PHP). Si vous spécifiez un sous-dossier ici (juste le nom, pas de «/» en tête),AWS OpsWorks Stacks l'ajoute au chemin de dossier standard. La version 2 de Simple PHPApp devrait entrer`/srv/www/web`, alors définissez Document root sur**`web`**.

   Data source type (Type de source de données)

   Ce paramètre associe un serveur de base de données à l'application. L'exemple utilise l'instance MySQL que vous avez créée à l'étape précédente. Définissez donc le type de source de données sur OpsWorks et l'instance de base de données sur l'instance que vous

avez créée à l'étape précédente, db-master1 (mysql). Laissez le champ Nom de la base de données vide ; AWS OpsWorks Stacks créera une base de données sur le serveur portant le nom abrégé de l'application, simplephpapp.

Puis, cliquez sur Save (Enregistrer) pour enregistrer la nouvelle configuration.



3. Démarrez l'instance MySQL.

Après avoir mis à jour une application, AWS OpsWorks Stacks déploie automatiquement la nouvelle version de l'application sur toutes les nouvelles instances de serveur d'applications lorsque vous les démarrez. Cependant, AWS OpsWorks Stacks ne déploie pas automatiquement la nouvelle version de l'application sur les instances de serveur existantes ; vous devez le faire manuellement, comme décrit dansÉtape 2.4 : Créer et déployer une application - Chef 11. Vous pouvez déployer la version mise à jour de Simple PHPApp dès maintenant, mais pour cet exemple, il vaut mieux attendre un peu.

Une courte digression : attributs des livres de cuisine, des recettes et des piles AWS OpsWorks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous avez maintenant des serveurs d'applications et de bases de données, mais ils ne sont pas tout à fait prêts à être utilisés. Vous devez toujours configurer la base de données et les paramètres de connexion de l'application. AWS OpsWorks Stacks ne gère pas ces tâches automatiquement, mais prend en charge les livres de recettes, les recettes et les attributs dynamiques de Chef. Vous pouvez implémenter deux recettes, l'une pour configurer la base de données et l'autre pour configurer les paramètres de connexion de l'application, et demander à AWS OpsWorks Stacks de les exécuter pour vous.

Le livre de recettes phpapp, qui contient les recettes nécessaires, est déjà implémenté et prêt à être utilisé ; vous pouvez juste passer à Étape 3.3 : Ajoutez les livres de recettes personnalisés à MyStack si vous préférez. Si vous souhaitez en savoir plus, cette section fournit quelques informations sur les livres de recettes et sur les recettes, et décrit comment les recettes fonctionnent. Pour consulter le livre de recettes, accédez au livre de recettes phpapp.

Rubriques

- Recettes et attributs
- Configurer la base de données
- Connecter l'application à la base de données

Recettes et attributs

Une recette Chef est essentiellement une application Ruby spécialisée qui effectue des tâches sur une instance, telles que l'installation de packages, la création de fichiers de configuration, l'exécution de commandes shell, etc. Les groupes de recettes connexes sont organisés en livres de recettes, qui contiennent également les fichiers de prise en charge tels que les modèles pour la création des fichiers de configuration.

AWS OpsWorks Stacks propose un ensemble de livres de recettes qui prennent en charge les couches intégrées. Vous pouvez aussi créer des livres de cuisine personnalisés avec vos propres recettes pour effectuer des tâches personnalisées sur vos instances. Cette rubrique fournit une brève introduction aux recettes et montre comment les utiliser pour configurer la base de données et les paramètres de connexion de l'application. Pour plus d'informations sur les livres de recettes et les recettes, consultez Livres de recettes et recettes ou Personnalisation des piles AWS OpsWorks.

Les recettes dépendent généralement des attributs Chef pour les données en entrée :

- Certains attributs sont définis par Chef et fournissent des informations de base sur l'instance, telles que le système d'exploitation.
- AWS OpsWorks Stacks définit un ensemble d'attributs contenant des informations sur la pile, telles que les configurations des couches, et sur les applications déployées, telles que le référentiel d'applications.

  Vous pouvez ajouter des attributs personnalisés à cet ensemble en affectant le JSON personnalisé à la pile ou au déploiement.
- Vos livres de recettes peuvent aussi définir des attributs, qui sont spécifiques au livre de recettes.

  Les attributs du livre de recettes phpapp sont définis dans `attributes/default.rb`.

Pour une liste complète des attributs de AWS OpsWorks Stacks, reportez-vous aux sections Attributs de déploiement et de configuration de pile : Linux etAttributs des livres de recettes intégrés. Pour de plus amples informations, veuillez consulter Remplacement des attributs.

Les attributs sont organisés selon une structure hiérarchique, laquelle peut être représentée sous forme d'un objet JSON.

Vous intégrez ces données dans votre application en utilisant la syntaxe nœud Chef, comme suit :

```
[:deploy][:simplephpapp][:database][:username]
```

Le nœud `deploy` possède un seul nœud d'application, `simplephpapp`, qui contient les informations sur la base de données de l'application, le référentiel Git, etc. L'exemple représente la valeur du nom d'utilisateur de la base de données, qui se résout en `root`.

Configurer la base de données

Les recettes de configuration intégrées à la couche MySQL créent automatiquement une base de données pour l'application nommée avec le nom abrégé de l'application. Dans cet exemple, vous avez déjà une base de données nommée simplephpapp. Cependant, vous devez terminer la configuration en créant une table pour que l'application stocke ses données. Vous pouvez créer le tableau manuellement, mais une meilleure approche consiste à implémenter une recette personnalisée pour gérer la tâche et à laisser AWS OpsWorks Stacks l'exécuter pour vous. Cette section décrit comment la recette, `dbsetup.rb`, est mise en œuvre. La procédure permettant à AWS OpsWorks Stacks d'exécuter la recette est décrite plus loin.

Pour consulter la recette dans le référentiel, accédez à dbsetup.rb. L'exemple suivant montre le code d'`dbsetup.rb`.

`execute` est une ressource Chef qui exécute une commande spécifiée. Dans ce cas, il s'agit d'une commande MySQL qui crée une table. La directive `not_if` garantit que la commande ne s'exécute pas si la table spécifiée existe déjà. Pour plus d'informations sur les ressources Chef, consultez À propos des ressources et des fournisseurs.

La recette insère les valeurs d'attribut dans la chaîne de commande, à l'aide de la syntaxe nœud discutée plus tôt. Par exemple, l'instruction suivante insère le nom d'utilisateur de la base de données.

```
#{deploy[:database][:username]}
```

Intéressons-nous de plus prés à ce code quelque peu énigmatique :

- Pour chaque itération, `deploy` est défini sur le nœud en cours de l'application et, par conséquent, se résout en `[:deploy][:app_name]`. Pour cet exemple, il est résolu en `[:deploy][:simplephpapp]`.
- En utilisant les valeurs d'attribut de déploiement illustrées plus tôt, le nœud entier se résout en `root`.
- Vous enveloppez le nœud dans #{ } pour l'insérer en une chaîne.

La plupart des autres nœuds se résolvent d'une manière similaire. L'exception est `#{node[:phpapp][:dbtable]}`, qui est défini par le fichier d'attributs du livre de recettes personnalisé et se résout en nom de la table, `urler`. La commande réelle qui s'exécute sur l'instance MySQL est donc :

```
"/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))'
"
```

Cette commande crée une table nommée `urler` avec les champs id, author et message, à l'aide des informations d'identification et du nom de base de données des attributs de déploiement.

Connecter l'application à la base de données

La deuxième pièce du puzzle est l'application, qui a besoin des informations de connexion telles que le mot de passe de la base de données pour accéder à la table. Simple n'a PHPApp effectivement qu'un seul fichier de travail`app.php`, `index.php` il ne fait que le charger`app.php`.

`app.php` comprend `db-connect.php`, qui gère la connexion de la base de données, mais ce fichier n'est pas dans le référentiel. Vous ne pouvez pas créer `db-connect.php` à l'avance, car il définit la base de données en fonction de l'instance particulière. Au lieu de cela, la recette `appsetup.rb` génère `db-connect.php` à l'aide des données de connexion des attributs de déploiement.

Pour consulter la recette dans le référentiel, accédez à appsetup.rb. L'exemple suivant montre le code d'`appsetup.rb`.

Par exemple`dbsetup.rb`, `appsetup.rb` itère sur les applications du nœud, encore une fois, une `deploy` simple application PHP. Il exécute un bloc de code avec une ressource `script` et une ressource `template`.

La `script` ressource installe Composer, un gestionnaire de dépendances pour les applications PHP. Il exécute ensuite la commande `install` de Composer pour installer les dépendances de l'exemple d'application sur le répertoire racine de l'application.

La ressource `template` génère `db-connect.php` et le met dans `/srv/www/simplephpapp/current`. Remarques :

- La recette utilise une instruction conditionnelle pour spécifier le propriétaire du fichier, lequel dépend du système d'exploitation de l'instance.
- La directive `only_if` demande à Chef de ne générer le modèle que si le répertoire spécifié existe.

Une ressource `template` fonctionne sur un modèle qui a pour l'essentiel les mêmes contenu et structure que le fichier associé, mais qui inclut des espaces réservés pour les différentes valeurs de données. Le paramètre `source` spécifie le modèle, `db-connect.php.erb`, qui se trouve dans le répertoire phpapp `templates/default` du livre de recettes phpapp et contient les éléments suivants :

Lorsque le Chef traite le modèle, il remplace les espaces réservés `<%= =>` par la valeur des variables correspondantes de la ressource modèle, qui proviennent à leur tour des attributs de déploiement. Le fichier généré se présente donc ainsi :

Étape 3.3 : Ajoutez les livres de recettes personnalisés à MyStack

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous stockez les livres de recettes personnalisés dans un référentiel, tout comme les applications. Chaque pile peut avoir un référentiel qui contient un ensemble de livres de recettes personnalisés. Vous demandez ensuite à AWS OpsWorks Stacks d'installer vos livres de recettes personnalisés sur les instances de la pile.

1. Cliquez sur Stack (Pile) dans le panneau de navigation pour afficher la page de la pile actuelle.
2. Cliquez sur Stack Settings (Paramètres de pile), puis sur Edit (Modifier).
3. Modifiez la configuration de la pile comme suit :

    - Utilisez des livres de cuisine Chef personnalisés — Oui

- Type de dépôt — Git
- URL du dépôt — **`git://github.com/amazonwebservices/opsworks-example-cookbooks.git`**

4. Cliquez sur Save (Enregistrer) pour mettre à jour la configuration de la pile.



AWS OpsWorks Stacks installe ensuite le contenu de votre référentiel de livres de recettes sur toutes les instances de la pile. Si vous créez de nouvelles instances, AWS OpsWorks Stacks installe automatiquement le référentiel de livres de recettes.

> **Note**
>
> Si vous devez mettre à jour l'un de vos livres de recettes ou en ajouter de nouveaux au référentiel, vous pouvez le faire sans toucher aux paramètres de la pile. AWS OpsWorks Stacks installera automatiquement les livres de recettes mis à jour sur toutes les nouvelles instances. Cependant, AWS OpsWorks Stacks n'installe pas automatiquement les livres de recettes mis à jour sur les instances en ligne de la pile. Vous devez explicitement demander à AWS OpsWorks Stacks de mettre à jour les livres de recettes en exécutant la commande `Update Cookbooks` stack. Pour de plus amples informations, veuillez consulter Exécution des commandes de pile.

Étape 3.4 : Exécuter les recettes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une fois que vous avez votre livre de recettes personnalisé, vous devez exécuter les recettes sur les instances appropriées. Vous pouvez les exécuter manuellement. Cependant, les recettes doivent généralement être exécutées sur des points prévisibles du cycle de vie d'une instance, tels qu'après l'amorçage de l'instance ou le moment où une application est déployée. Cette section décrit une approche beaucoup plus simple : demandez à AWS OpsWorks Stacks de les exécuter automatiquement pour vous au moment opportun.

AWS OpsWorks Stacks prend en charge un ensemble d'événements du cycle de vie qui simplifient les recettes d'exécution. Par exemple, l'événement Setup se produit lorsqu'une instance finit le démarrage et l'événement Deploy se produit lorsque vous déployez une application. Chaque couche possède un ensemble de recettes intégrées associées à chaque événement du cycle de vie. Lorsqu'un événement du cycle de vie se produit sur une instance, l'agent exécute les recettes associées pour chacune des couches de l'instance. Pour que AWS OpsWorks Stacks exécute automatiquement une recette personnalisée, ajoutez-la à l'événement du cycle de vie approprié sur la couche appropriée et l'agent exécutera la recette une fois les recettes intégrées terminées.

Pour cet exemple, vous devez exécuter deux recettes, `dbsetup.rb` sur l'instance My SQLinstance et `appsetup.rb` sur l'instance PHP App Server.

> **Note**
>
> Vous spécifiez les recettes sur la console en utilisant le *`recipe_name`* format *`cookbook_name`* : :, dans lequel l'*`recipe_name`*extension .rb n'est pas incluse. Par exemple, vous faites référence à `dbsetup.rb` sous la forme **`phpapp::dbsetup`**.

Pour assigner les recettes personnalisées aux événements de cycle de vie

1. Sur la page Couches, pour MySQL, cliquez sur Recettes, puis sur Modifier.
2. Dans la section Custom Chef recipes (Recettes Chef personnalisées), entrez **`phpapp::dbsetup`** pour Deploy.



3. Cliquez sur l'icône + pour affecter la recette à l'événement et cliquez sur Save (Enregistrer) pour enregistrer la nouvelle configuration de la couche.
4. Retournez à la page des couches et répétez la procédure d'attribution **`phpapp::appsetup`** à l'événement Deploy de la couche PHP App Server.

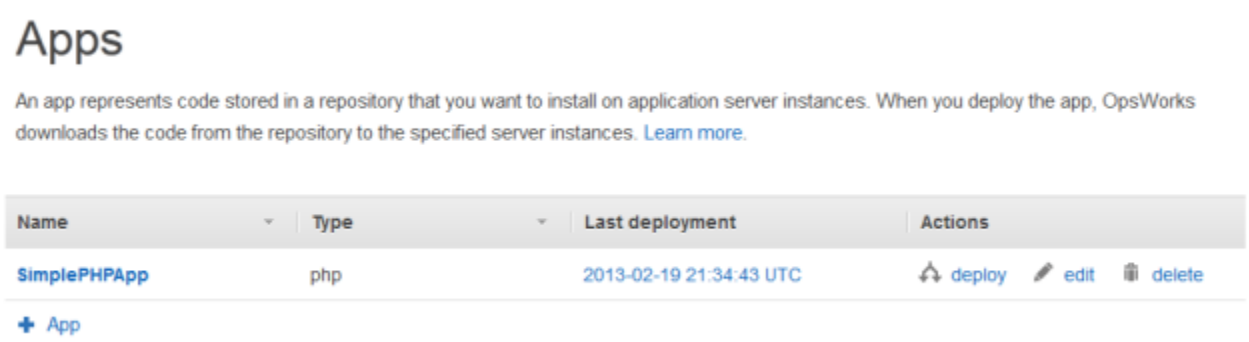
Étape 3.5 : Déploiement de SimplePHPApp, version 2

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

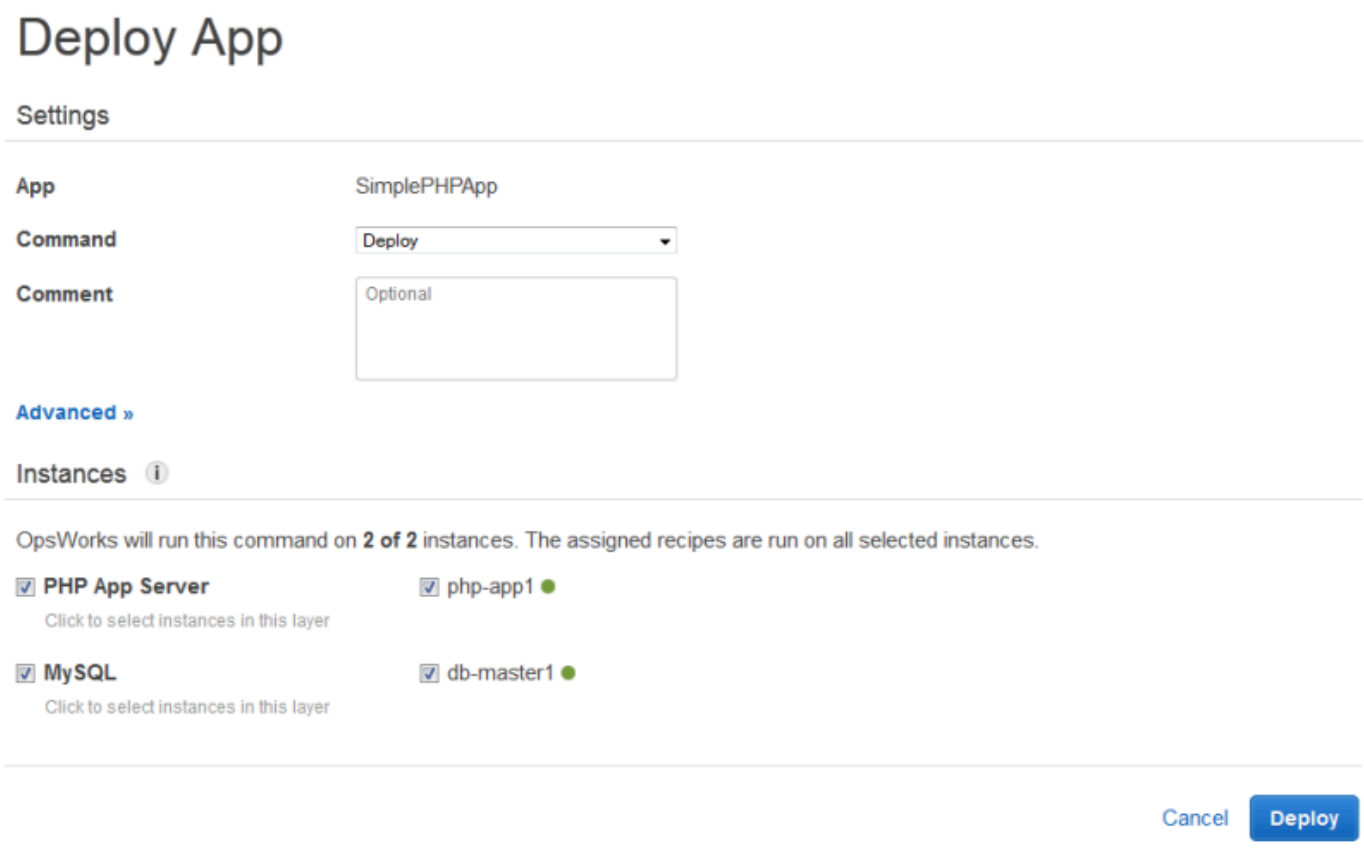
La dernière étape consiste à déployer la nouvelle version de SimplePHPApp.

Pour déployer Simple PHPApp

1. Sur la page Applications, cliquez sur Déployer dans Actions de PHPApp l'application Simple.



2. Acceptez les valeurs par défaut et cliquez sur Deploy (Déployer).



Lorsque vous cliquez sur Deploy (Déployer) sur la page Deploy App (Déployer l'application), vous déclenchez un événement de cycle de vie Deploy, qui demande aux agents d'exécuter leurs recettes Deploy. Par défaut, vous déclenchez l'événement sur l'ensemble des instances de la pile. Les recettes de déploiement intégrées déploient l'application uniquement sur les instances appropriées au type d'application, les instances de PHP App Server dans ce cas. Cependant, il est souvent utile de déclencher l'événement Deploy sur d'autres instances, pour leur permettre de répondre au déploiement d'applications. Dans ce cas, vous souhaitez également déclencher Deploy sur l'instance MySQL pour configurer la base de données.

Remarques :

- L'agent de l'instance PHP App Server exécute la recette intégrée de la couche`appsetup.rb`, suivie de celle qui configure la connexion à la base de données de l'application.
- L'agent de l'instance MySQL n'installe rien, mais il s'exécute `dbsetup.rb` pour créer la table des urler.

Lorsque le déploiement est terminé, Status (Statut) devient successful (succès) sur la page Deployment (Déploiement).

Étape 3.6 : Exécuter en toute simplicité PHPApp

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une fois que l'état du déploiement est passé à Successful, vous pouvez exécuter la nouvelle PHPApp version Simple comme suit.

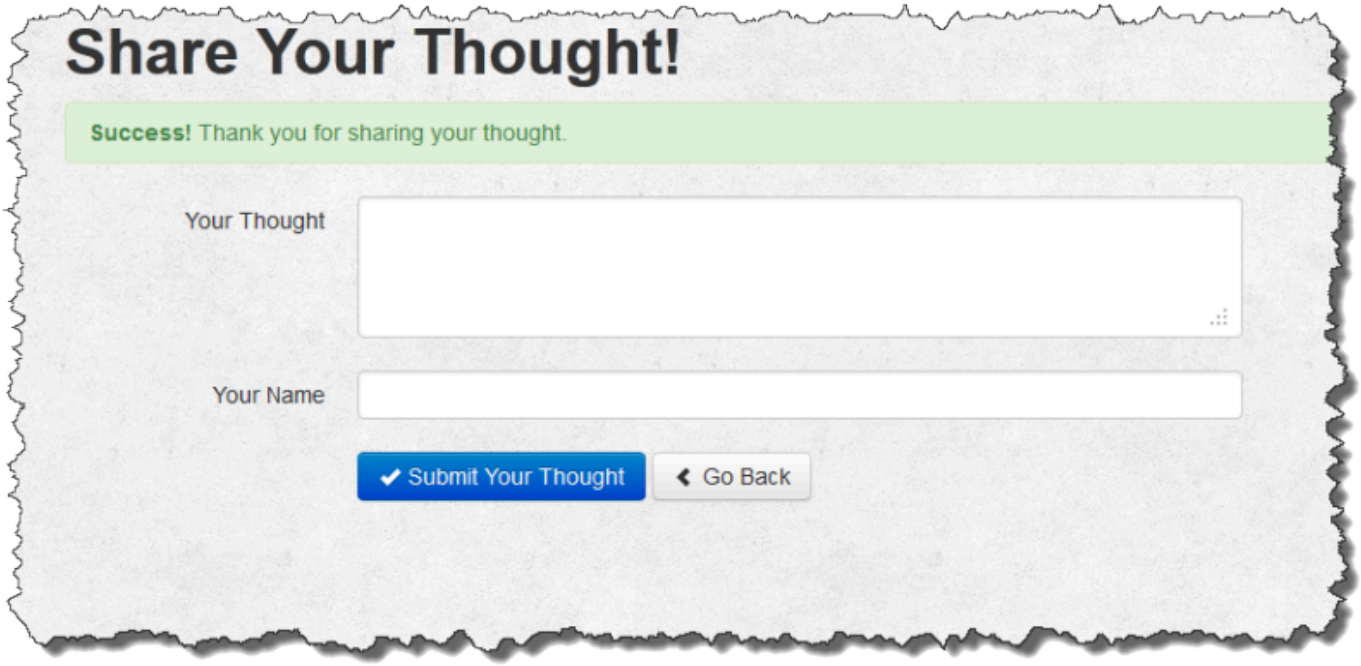
Pour exécuter Simple PHPApp

1. Sur la page Instances, cliquez sur l'adresse IP publique dans la ligne php-app1.

   Vous devez voir la page suivante dans votre navigateur.



2. Cliquez sur Share Your Thought et tapez quelque chose comme **`Hello world!`** pour Your Thought et votre nom pour Your Name. Puis, cliquez sur Submit Your Thought pour ajouter le message à la base de données.

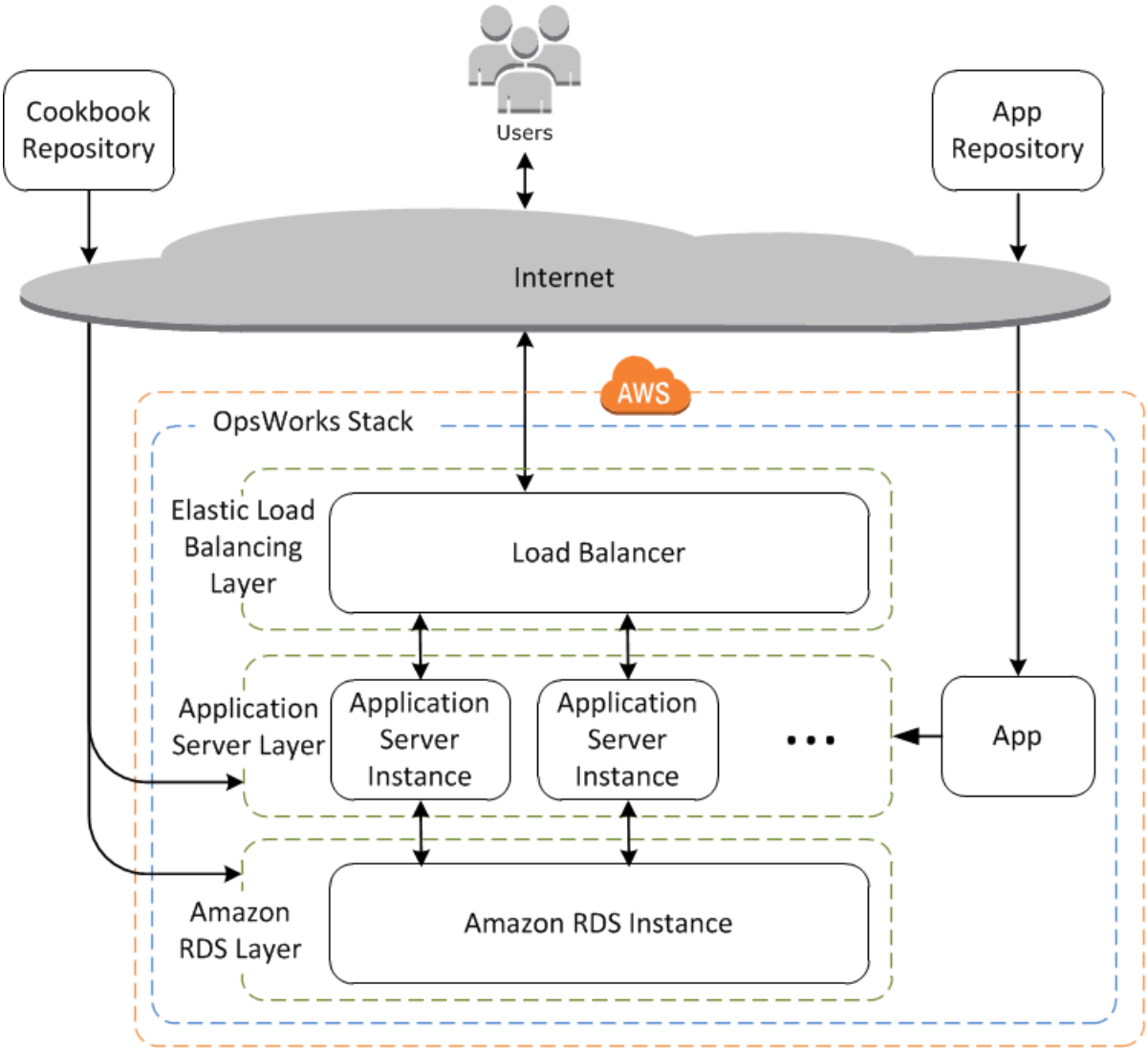


3. Cliquez sur Go Back pour afficher tous les messages de la base de données.

Étape 4 : Diminution MyStack

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

MyStack ne possède actuellement qu'un seul serveur d'applications. Une pile en production aura sans doute besoin de plusieurs serveurs d'applications pour gérer le trafic entrant et d'un équilibreur de charge pour répartir le trafic entrant de façon égale entre les serveurs d'applications. L'architecture ressemblera à ce qui suit :



AWS OpsWorks Stacks permet de redimensionner facilement les piles. Cette section décrit les principes de base de la mise à l'échelle d'une pile en ajoutant une deuxième instance PHP App Server 24 h/24, 7 j/7 MyStack et en plaçant les deux instances derrière un équilibreur de charge Elastic Load Balancing. Vous pouvez facilement étendre la procédure pour ajouter un nombre arbitraire d'instances 24 heures sur 24, 7 jours sur 7, ou vous pouvez utiliser des instances basées sur le temps ou sur la charge pour que AWS OpsWorks Stacks redimensionne automatiquement votre stack. Pour de plus amples informations, veuillez consulter Gestion de la charge avec des instances basées sur le temps et sur la charge.

Étape 4.1 : Ajouter un équilibreur de charge

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Elastic Load Balancing est un service AWS qui distribue automatiquement le trafic applicatif entrant sur plusieurs EC2 instances Amazon. Outre la distribution du trafic, Elastic Load Balancing effectue les tâches suivantes :

- Détecte les EC2 instances Amazon défectueuses.

  Il redirige le trafic vers les instances saines restantes en attendant que les instances défectueuses soient restaurées.
- Dimensionne automatiquement la capacité de traitement des demandes en réponse au trafic entrant

> **Note**
>
> Un équilibreur de charge peut obéir à deux objectifs. L'objectif manifeste consiste à équilibrer la charge sur vos serveurs d'applications. En outre, la plupart des sites préfèrent isoler leurs serveurs d'applications et bases de données d'un accès direct de l'utilisateur. Avec AWS OpsWorks Stacks, vous pouvez le faire en exécutant votre stack dans un cloud privé virtuel (VPC) avec un sous-réseau public et privé, comme suit.
>
> - Placez les serveurs d'applications et la base de données dans le sous-réseau privé, où ils sont accessibles par d'autres instances du VPC, mais pas par les utilisateurs.
> - Dirigez le trafic utilisateur vers un équilibreur de charge du sous-réseau public, qui achemine alors le trafic vers les serveurs d'applications du sous-réseau privé, puis renvoie les réponses aux utilisateurs.

Pour de plus amples informations, veuillez consulter Running a Stack in a VPC. Pour un AWS CloudFormation modèle qui étend l'exemple de cette procédure pas à pas pour qu'il s'exécute dans un VPC, téléchargez `OpsWorksVPCtemplates.zip` le fichier.

Bien qu'Elastic Load Balancing soit souvent appelé couche, il fonctionne un peu différemment des autres couches intégrées. Au lieu de créer une couche et d'y ajouter des instances, vous créez un équilibreur de charge Elastic Load Balancing à l'aide de la EC2 console Amazon, puis vous l'attachez à l'une de vos couches existantes, généralement une couche de serveur d'applications. AWS OpsWorks Stacks enregistre ensuite les instances existantes de la couche auprès du service et ajoute automatiquement les nouvelles instances. La procédure suivante décrit comment ajouter un équilibreur de charge à MyStack la couche PHP App Server.

> **Note**
>
> AWS OpsWorks Stacks ne prend pas en charge Application Load Balancer. Vous ne pouvez utiliser Classic Load Balancer qu'avec AWS OpsWorks Stacks.

Pour attacher un équilibreur de charge à la couche PHP App Server

1. Utilisez la EC2 console Amazon pour créer un nouvel équilibreur de charge pour MyStack. Les détails varient selon que votre compte est compatible avec EC2 Classic. Pour plus d'informations, consultez Mise en route avec Elastic Load Balancing. Lorsque vous exécutez l'Assistant Créer un équilibreur de charge, configurez l'équilibreur de charge comme suit :

   Définir un équilibreur de charge

   Attribuez à l'équilibreur de charge un nom facilement reconnaissable, tel que PHP-LB, afin de le localiser plus facilement dans la AWS OpsWorks console Stacks. Puis, choisissez Continuer pour accepter les valeurs par défaut des paramètres restants.

   Si vous sélectionnez un VPC avec un ou plusieurs sous-réseaux à partir du menu Créer un équilibreur dans, vous devez sélectionner un sous-réseau pour chaque zone de disponibilité où vous voulez que le trafic soit acheminé par votre équilibreur de charge.

Attribuer les groupes de sécurité

Si votre compte prend en charge le VPC par défaut, l'Assistant affiche cette page pour déterminer le groupe de sécurité de l'équilibreur de charge. Cette page n'est pas affichée pour EC2 Classic.

Pour cette procédure pas à pas, choisissez groupe de sécurité par défaut du VPC.

Configurer les paramètres de sécurité

Si vous avez choisi HTTPS comme Protocole de l'équilibreur de charge sur la page Définir un équilibreur de charge, configurez les paramètres de certificat, de chiffrement et de protocole SSL sur cette page. Pour cette procédure pas à pas, acceptez les valeurs par défaut, puis sélectionnez Configurer la vérification de l'état.

Configurer la vérification de l'état

Définissez le chemin d'accès ping sur **`/`** et acceptez les valeurs par défaut des paramètres restants.

Ajouter des EC2 instances

Choisissez Continuer ; AWS OpsWorks Stacks enregistre automatiquement les instances auprès de l'équilibreur de charge.

Ajouter des balises

Ajoutez des balises pour vous aider à trouver. Chaque balise est une paire clé/valeur ; par exemple, vous pouvez spécifier **`Description`** comme clé et **`Test LB`** comme valeur dans le cadre de la procédure pas à pas.

Vérification

Passez en revue votre choix, puis choisissez Créer et Fermer. Cela démarre l'équilibreur de charge.

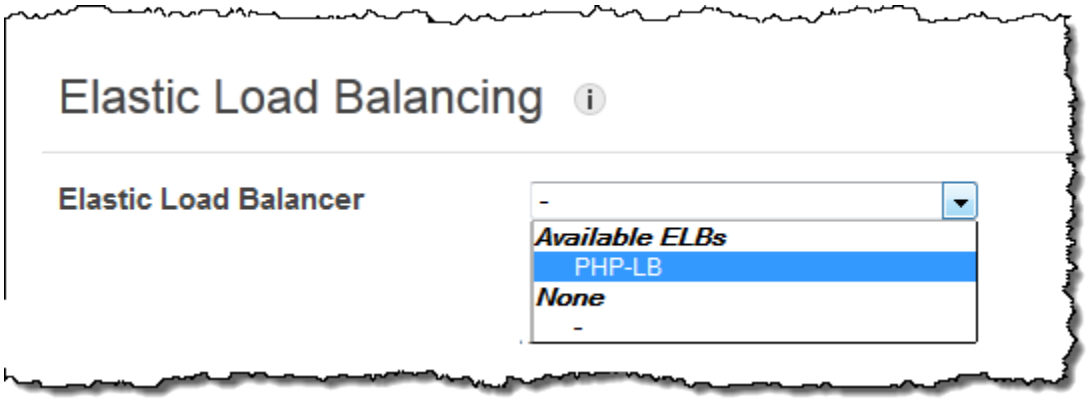
2. Si votre compte prend en charge les VPC par défaut, après que vous avez démarré l'équilibreur de charge, vous devez vous assurer que son groupe de sécurité a les règles de trafic entrant appropriées. La règle par défaut n'accepte pas de trafic entrant.

   1. Choisissez Security Groups dans le volet EC2 de navigation Amazon.
   2. Sélectionnez Groupe de sécurité par défaut du VPC.
   3. Choisissez Modifier sous l'onglet Entrant.

4. Pour cette procédure pas à pas, définissez Source sur N'importe où, ce qui demande à l'équilibreur de charge d'accepter le trafic entrant à partir de n'importe quelle adresse IP.

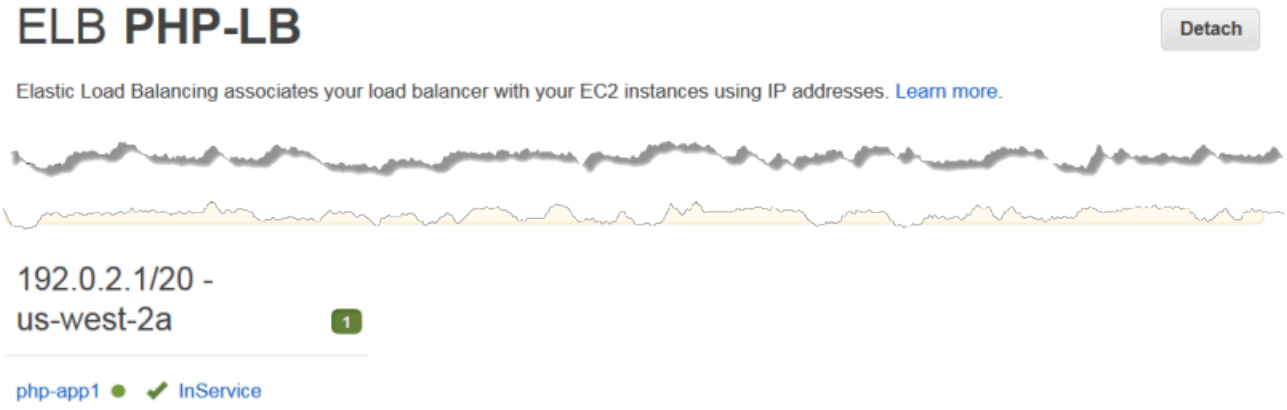
3. Retournez à la console AWS OpsWorks Stacks. Sur la page Couches, choisissez le lien Réseau de la couche, puis choisissez Modifier.

4. Sous Elastic Load Balancing, sélectionnez l'équilibreur de charge que vous avez créé à l'étape 1, puis Enregistrer.



Une fois que vous avez attaché l'équilibreur de charge à la couche, AWS OpsWorks Stacks enregistre automatiquement les instances actuelles de la couche et en ajoute de nouvelles au fur et à mesure de leur mise en ligne.

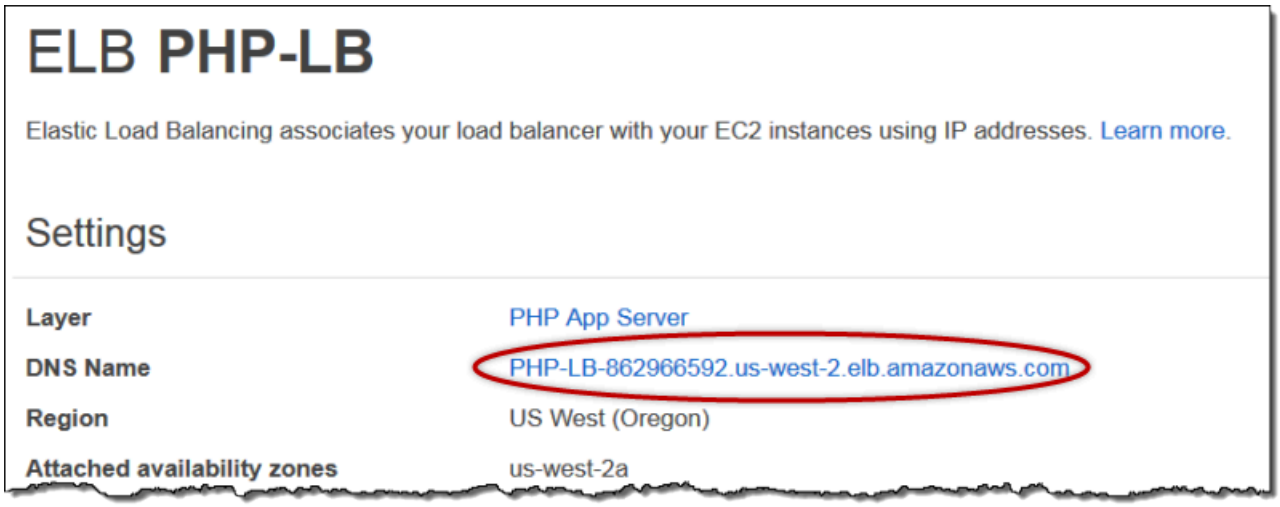
5. Sur la page Couches, cliquez sur le nom de l'équilibreur de charge pour ouvrir la page des détails. Lorsque l'enregistrement est terminé et que l'instance passe un contrôle de santé, AWS OpsWorks Stacks affiche une coche verte à côté de l'instance sur la page de l'équilibreur de charge.



Vous pouvez désormais exécuter Simple PHPApp en envoyant une demande à l'équilibreur de charge.

Pour exécuter Simple PHPApp via l'équilibreur de charge

1. Ouvrez à nouveau la page des détails de l'équilibreur de charge, s'il n'est pas déjà ouvert.
2. Sur la page des propriétés, vérifiez l'état de santé de l'instance et cliquez sur le nom DNS de l'équilibreur de charge pour exécuter Simple. PHPApp L'équilibreur de charge transmet la demande à l'instance PHP App Server et renvoie la réponse, qui doit être exactement la même que celle que vous obtenez lorsque vous cliquez sur l'adresse IP publique de l'instance PHP App Server.



> **Note**
>
> AWS OpsWorks Stacks prend également en charge l'équilibreur de HAProxy charge, ce qui peut présenter des avantages pour certaines applications. Pour de plus amples informations, veuillez consulter HAProxy AWS OpsWorks Couche Stacks.

Étape 4.2 : Ajouter des instances de serveur d'applications PHP

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Maintenant que l'équilibreur de charge est en place, vous pouvez augmenter la taille de la pile en ajoutant plus d'instances à la couche PHP App Server. De votre point de vue, l'opération est transparente. Chaque fois qu'une nouvelle instance de PHP App Server est mise en ligne, AWS OpsWorks Stacks l'enregistre automatiquement auprès de l'équilibreur de charge et déploie SimplePHPApp, afin que le serveur puisse immédiatement commencer à gérer le trafic entrant. Par souci de concision, cette rubrique explique comment ajouter une instance supplémentaire de PHP App Server, mais vous pouvez utiliser la même approche pour en ajouter autant que vous le souhaitez.

Pour ajouter une autre instance à la couche PHP App Server

1. Sur la page Instances, cliquez sur + Instance sous PHP App Server.
2. Acceptez les paramètres par défaut et cliquez sur Add Instance (Ajouter une instance).
3. Cliquez sur start (démarrer) pour démarrer l'instance.

Étape 4.3 : Surveiller MyStack

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

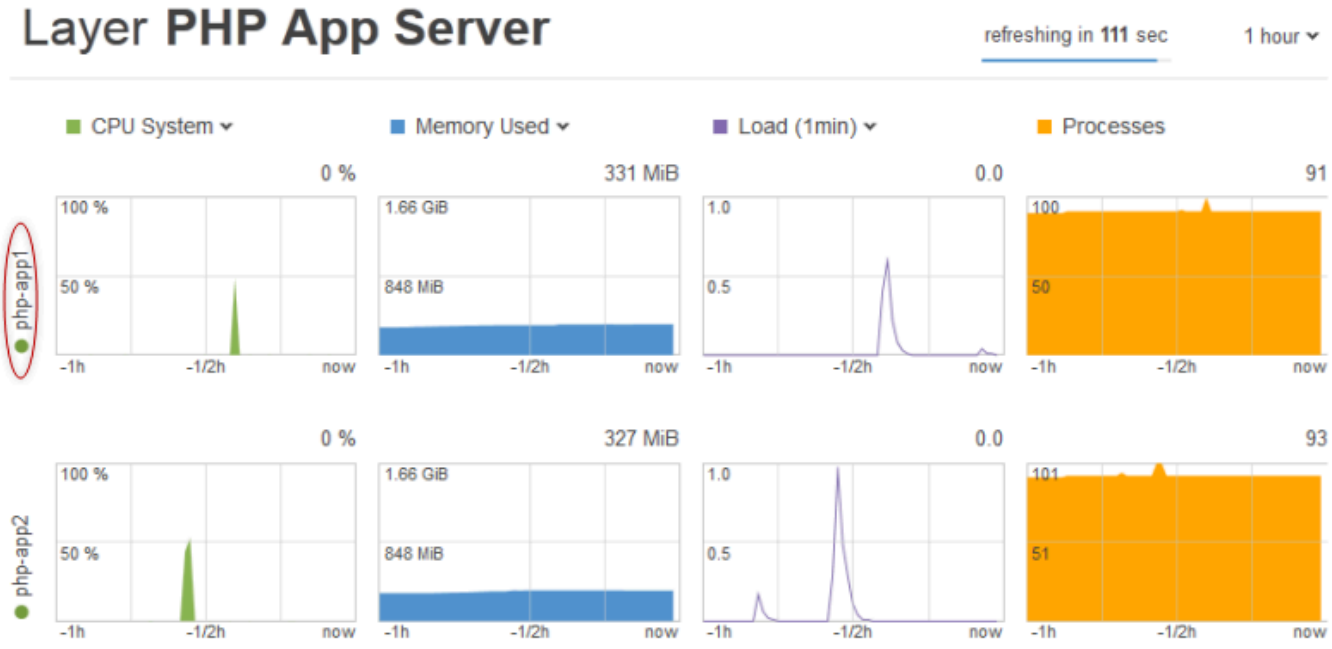
AWS OpsWorks Stacks utilise Amazon CloudWatch pour fournir les statistiques d'une pile et les résume pour vous faciliter la tâche sur la page de surveillance. Vous pouvez afficher les métriques pour l'ensemble de la pile, une couche donnée ou une instance spécifiée.

À surveiller MyStack

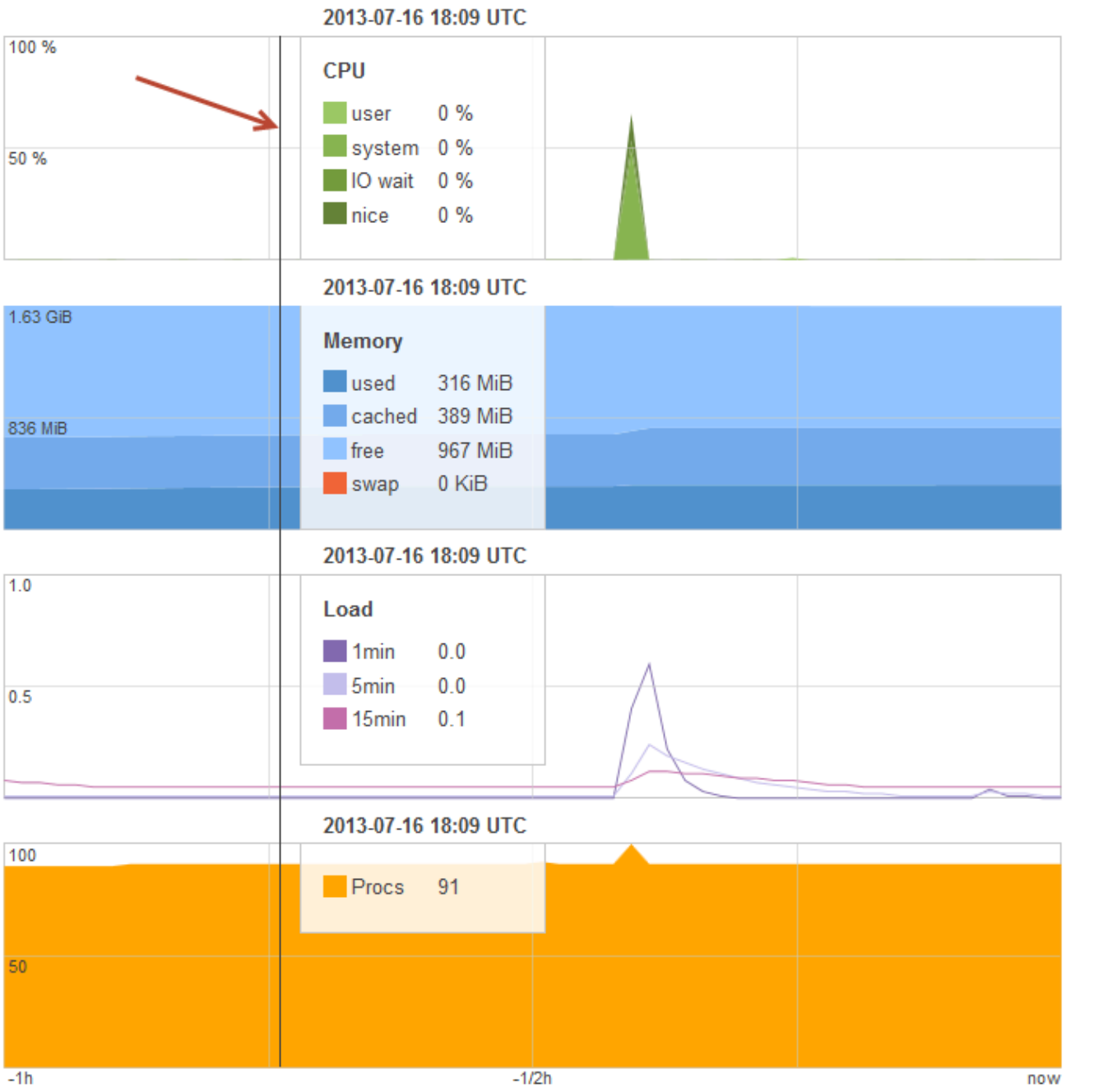
1. Dans le panneau de navigation, cliquez sur Monitoring (Surveillance), ce qui affiche un ensemble de graphes avec les métriques moyennes de chaque couche. Vous pouvez utiliser les menus de CPU System (UC système), Memory Used (Mémoire utilisée) et Load (Charge) pour afficher les différentes métriques liées.



2. Cliquez sur PHP App Server (Serveur d'applications PHP) pour afficher les métriques de chacune des instances de la couche.



3. Cliquez sur php-app1 pour afficher les métriques de cette instance. Vous pouvez voir les métriques associées à n'importe quel point dans le temps en déplaçant le curseur.


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 prend également en charge le serveur de surveillance Ganglia, ce qui peut présenter des avantages pour certaines applications. Pour de plus amples informations, veuillez consulter Couche ganglionnaire.

Étape 5 : Supprimer MyStack

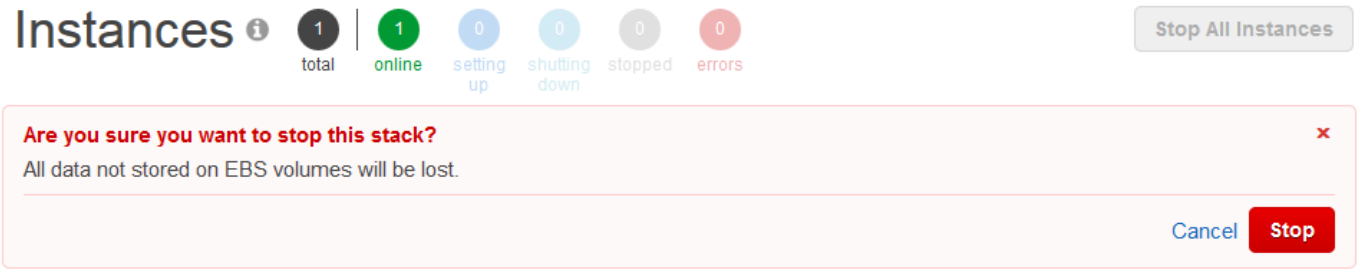
> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Dès que vous commencez à utiliser des ressources AWS telles que des EC2 instances Amazon, vous êtes facturé en fonction de votre utilisation. Si vous avez terminé, vous devez arrêter les instances afin de ne pas vous exposer à des frais non désirés. Si vous n'avez plus besoin de la pile, vous pouvez la supprimer.

Pour supprimer MyStack

1. Arrêter toutes les instances

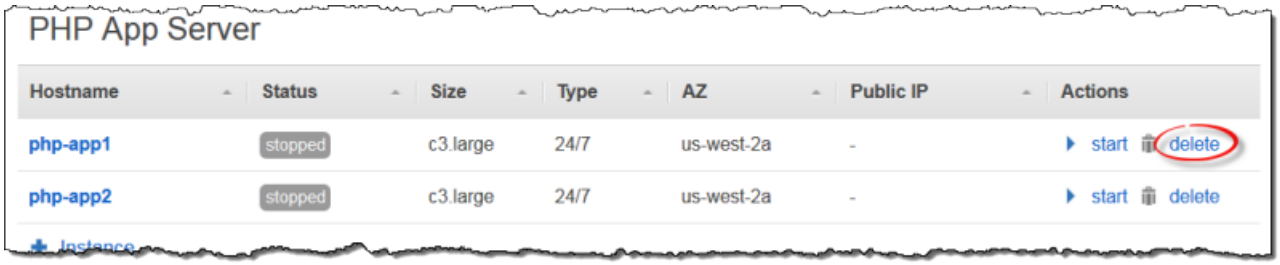
   Sur la page Instances, cliquez sur Stop All Instances (Arrêter toutes les instances) et cliquez sur Stop lorsqu'il vous est demandé de confirmer l'opération.



Une fois que vous avez cliqué sur Stop, AWS OpsWorks Stacks met fin aux EC2 instances Amazon associées, mais pas aux ressources associées telles que les adresses IP Elastic ou les volumes Amazon EBS.

2. Supprimer toutes les instances

   L'arrêt de l'instance met simplement fin aux EC2 instances Amazon associées. Une fois que l'état des instances est déclaré comme arrêté, vous devez supprimer chaque instance. Dans la couche PHP App Server (Serveur d'applications PHP), cliquez sur delete (supprimer) dans la colonne Actions de l'instance php-app1.

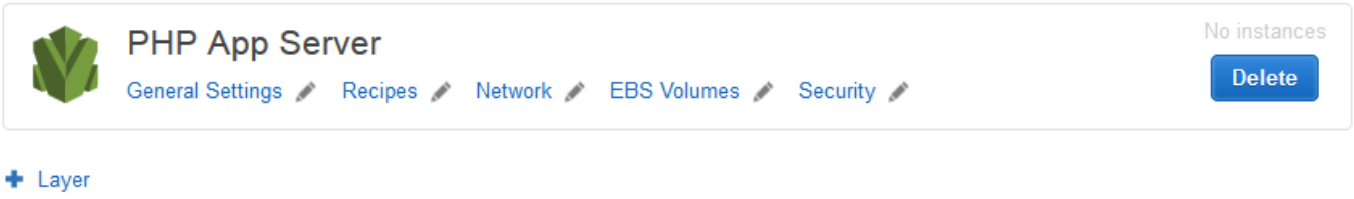


   AWS OpsWorks Stacks vous demande ensuite de confirmer la suppression et vous montre toutes les ressources dépendantes. Vous pouvez choisir de garder tout ou partie de ces ressources. Cet exemple n'a aucune ressource dépendante ; par conséquent, cliquez juste sur Delete (Supprimer).

   Répétez le processus pour php-app2 et l'instance MySQL, db-master1. Notez que db-master1 possède un volume Amazon Elastic Block Store associé, qui est sélectionné par défaut. Conservez la sélection pour supprimer le volume en même temps que l'instance.

3. Supprimez les couches.

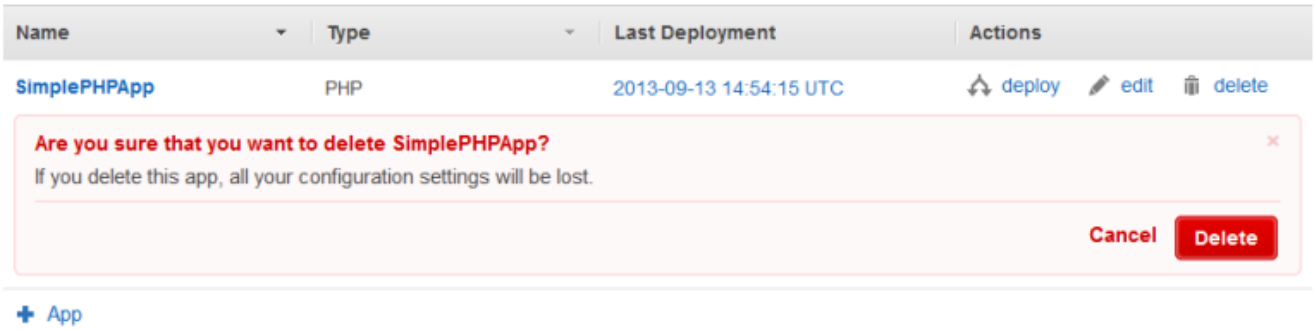
   Sur la page Layers (Couches), cliquez sur Delete (Supprimer), puis cliquez sur Delete (Supprimer) pour confirmer.



   Répétez l'opération pour la couche MySQL.

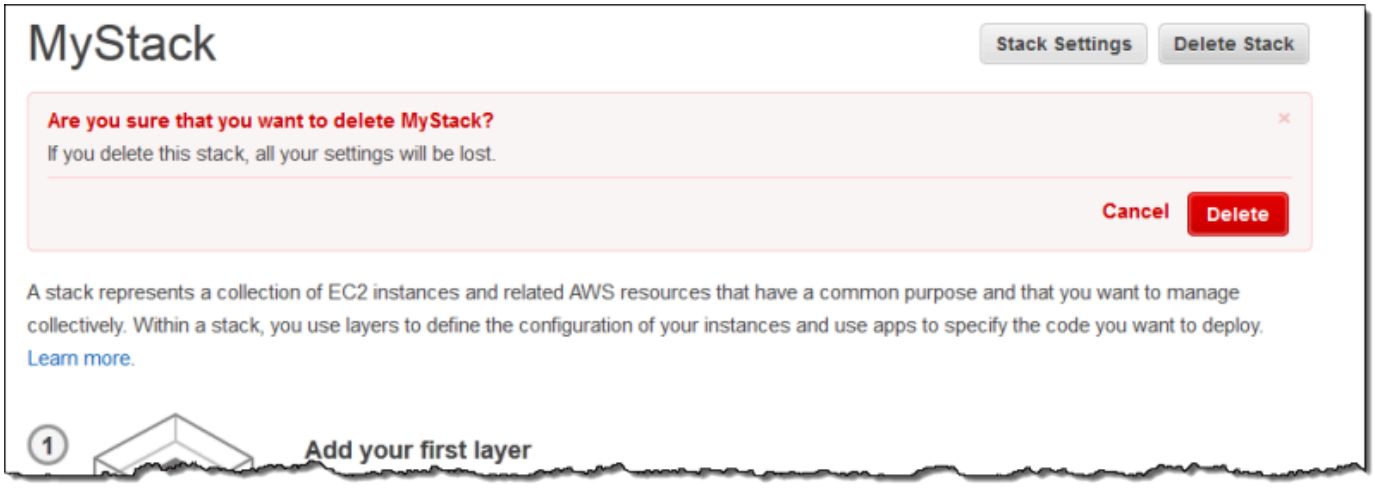
4. Supprimer l'application

   Sur la page Applications, cliquez sur Supprimer dans la colonne Actions de PHPApp l'application simple, puis sur Supprimer pour confirmer.



5. Supprimer MyStack

   Sur la page Stack (Pile), cliquez sur Delete Stack (Supprimer la pile), puis cliquez sur Delete (Supprimer) pour confirmer.



Vous avez terminé la procédure.

Création de votre première pile Node.js

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cet exemple explique comment créer une pile Linux qui prend en charge un serveur d'applications Node.js et comment déployer une application simple. La pile se compose des éléments suivants :

- Une couche de serveur d'applications Node.js avec deux instances
- Un équilibreur de charge Elastic Load Balancing pour distribuer le trafic vers les instances du serveur d'applications
- Une couche de service Amazon Relational Database Service (Amazon RDS) qui fournit une base de données principale

Rubriques

- Prérequis
- Implémentation de l'application
- Création du serveur de base de données et de l'équilibreur de charge
- Création de la pile
- Déploiement de l'application
- Et maintenant ?

Prérequis

Cette procédure pas à pas présume ce qui suit :

- Vous possédez un compte AWS et des connaissances de base sur l'utilisation de AWS OpsWorks Stacks.

  Si vous utilisez AWS OpsWorks Stacks ou AWS pour la première fois, découvrez les bases en suivant le didacticiel d'introduction dansMise en route des piles Linux Chef 11.
- Vous avez une compréhension de base de la façon d'implémenter une application Node.js.

  Si vous utilisez Node.js pour la première fois, découvrez les principes de base en suivant un didacticiel d'introduction, tel que Node: Up and Running.
- Vous avez déjà créé au moins une pile dans la région AWS que vous envisagez d'utiliser pour cet exemple.

Lorsque vous créez la première pile dans une région, AWS OpsWorks Stacks crée un groupe de sécurité Amazon Elastic Compute Cloud (Amazon EC2) pour chaque type de couche. Vous avez besoin de ces groupes de sécurité pour créer l'instance de base de données (DB) Amazon RDS. Si vous utilisez AWS OpsWorks Stacks pour la première fois, nous vous recommandons d'utiliser pour cet exemple la même région que celle que vous avez utilisée lorsque vous avez suivi le didacticiel dansMise en route des piles Linux Chef 11. Si vous souhaitez utiliser une nouvelle région, créez une pile dans la région ; la pile n'a pas besoin d'avoir de couches ou d'instances. Dès que vous créez la pile, AWS OpsWorks Stacks ajoute automatiquement un ensemble de groupes de sécurité à la région.

- Vous allez créer votre pile dans un VPC par défaut.

  Vous pouvez utiliser EC2 -Classic pour cette procédure pas à pas, mais certains détails seront légèrement différents. Par exemple, avec EC2 -Classic, vous spécifiez la zone de disponibilité (AZ) d'une instance au lieu de son sous-réseau.

- Votre utilisateur IAM dispose d'autorisations d'accès complètes à Stacks. AWS OpsWorks

  Pour des raisons de sécurité, nous vous recommandons vivement de ne pas utiliser les informations d'identification racine de votre compte pour cette procédure pas à pas. Créez plutôt un utilisateur avec des autorisations d'accès complètes à AWS OpsWorks Stacks et utilisez ces informations d'identification avec AWS OpsWorks Stacks. Pour de plus amples informations, veuillez consulter Création d'un utilisateur administrateur .

Implémentation de l'application

Cette procédure pas à pas utilise une simple application Express qui se connecte à l'instance de base de données Amazon RDS et répertorie les bases de données de l'instance.

Pour implémenter l'application, créez un répertoire nommé `nodedb` sur un emplacement approprié de votre station de travail, puis ajoutez-y les trois fichiers suivants.

Rubriques

- Descripteur du package
- Fichier de présentation
- Fichier de code

Descripteur du package

Pour créer le descripteur du package de l'application, ajoutez un fichier nommé `package.json` avec le contenu suivant dans le répertoire `nodedb`. `package.json` est obligatoire pour les applications Express et doit être situé dans le répertoire racine de l'application.

```
{
  "name": "Nodejs-DB",
  "description": "Node.js example application",
  "version": "0.0.1",
  "dependencies": {
    "express": "*",
    "ejs": "*",
    "mysql": "*"
  }
}
```

L'exemple `package.json` est assez rudimentaire. Il définit les attributs `name` et `version` requis, et répertorie les packages dépendants :

- `express` référence le package Express.
- `ejs` référence le package EJS, que l'application utilise pour insérer du texte dans un fichier de mise en page HTML.
- `mysql` référence le package node-mysql, que l'application utilise pour se connecter à l'instance RDS.

Pour plus d'informations sur les fichiers descripteurs du package, consultez package.json.

Fichier de présentation

Pour créer le fichier de présentation de l'application, ajoutez un répertoire `views` au répertoire `nodedb`, puis ajoutez au répertoire `views` un fichier nommé `index.html` avec le contenu suivant :

```
<!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>
```

Dans cet exemple, le fichier de mise en page est un simple document HTML qui affiche certaines données d'Amazon RDS. Chaque élément `<%= ... =>` représente la valeur d'une variable définie dans le fichier de code de l'application, que nous allons maintenant créer.

Fichier de code

Pour créer le fichier de code de l'application, ajoutez un fichier `server.js` au répertoire `nodedb` avec le contenu suivant.

> **Important**
>
> Avec AWS OpsWorks Stacks, le fichier de code principal d'une application Node.js doit être nommé `server.js` et se trouver dans le dossier racine de l'application.

```
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);
```

L'exemple affiche les informations de connexion de base de données ; en outre, il interroge également le serveur de base de données et affiche les bases de données du serveur. Vous pouvez facilement le généraliser pour interagir avec la base de données en fonction des besoins. Les notes suivantes se rapportent aux commentaires numérotés du code précédent.

[1] inclure les données de connexion de base de données

L'instruction `require` inclut les données de connexion de base de données. Comme décrit plus loin, lorsque vous attachez une instance de base de données à une application, AWS OpsWorks Stacks place les données de connexion dans un fichier nomméopsworks.js, qui ressemble à ce qui suit :

```
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` se trouve dans le répertoire `shared/config` de l'application `/srv/www/app_shortname/shared/config`. Cependant, AWS OpsWorks Stacks place un lien symbolique `opsworks.js` dans le répertoire racine de l'application, afin que vous puissiez inclure l'objet en utilisant simplement. `require 'opsworks'`

[2] Obtenir les données de connexion de base de données

Cet ensemble d'instructions affiche les données de connexion à partir d'`opsworks.js` en affectant les valeurs de l'objet db à un ensemble de propriétés `app.locals`, chacune d'elles étant mappée à l'un des éléments <%= ... %> du fichier `index.html`. Le document rendu remplace les éléments <%= ... %> par les valeurs de propriété correspondantes.

[3] Se connecter à l'instance Amazon RDS

L'exemple utilise `node-mysql` pour accéder à la base de données. Pour se connecter à la base de données, l'exemple crée un objet `connection` en transmettant les données de connexion à `createConnection`, puis appelle `connection.connect` pour établir la connexion.

[4] Interroger la base de données

Après avoir établi une connexion, l'exemple appelle `connection.query` pour interroger la base de données. Cet exemple interroge simplement les noms de base de données du serveur. `query` retourne un tableau d'objets `results`, un pour chaque base de données, avec le nom de base de données assigné à la propriété `Database`. L'exemple concatène les noms et les attributs, et les assigne à `app.locals.databases,`, qui affiche la liste dans la page HTML restituée.

Dans cet exemple, il existe cinq bases de données, la `nodeexampledb` base de données que vous avez spécifiée lors de la création de l'instance RDS et quatre autres créées automatiquement par Amazon RDS.

[5] Écouter les demandes entrantes

L'instruction finale écoute les demandes entrantes sur un port spécifié. Vous n'avez pas à spécifier une valeur de port explicite. Lorsque vous ajoutez l'application à votre pile, vous indiquez si l'application prend en charge les requêtes HTTP ou HTTPS. AWS OpsWorks Stacks définit ensuite la variable d'`PORT`environnement sur 80 (HTTP) ou 443 (HTTPS), et vous pouvez utiliser cette variable dans votre application.

Il est possible d'écouter sur d'autres ports, mais le groupe de sécurité intégré à la couche Node.js App Server, AWS- OpsWorks -NodeJS-App-Server, autorise le trafic utilisateur entrant uniquement vers les ports 80, 443 et 22 (SSH). Pour autoriser le trafic utilisateur entrant vers d'autres ports, créez un groupe de sécurité avec des règles de trafic entrant appropriées et attribuez-le à la couche Node.js App Server. Ne changez pas les règles de trafic entrant en modifiant le groupe de sécurité intégré. Chaque fois que vous créez une pile, AWS OpsWorks Stacks remplace les groupes de sécurité intégrés par les paramètres standard, de sorte que toutes les modifications que vous apportez seront perdues.

> **Note**
>
> Vous pouvez associer des variables d'environnement personnalisées à votre application lorsque vous créez ou mettez à jour l'application associée. Vous pouvez également passer les données à votre application en utilisant une recette JSON personnalisée et une recette personnalisée. Pour de plus amples informations, veuillez consulter Transmission de données aux applications.

Création du serveur de base de données et de l'équilibreur de charge

Cet exemple utilise le serveur de base de données Amazon RDS et les instances d'équilibreur de charge Elastic Load Balancing. Vous devez créer chaque instance séparément, puis l'intégrer à votre pile. Cette section explique comment créer les nouvelles instances de base de données et d'équilibreur de charge. Vous pouvez ensuite utiliser à la place les instances existantes, mais nous vous recommandons de lire la procédure afin de vous assurer que ces instances sont correctement configurées.

La suite décrit comment créer une instance de base de données RDS, ce qui suffit à cet exemple. Pour plus d'informations, veuillez consulter le Guide d'utilisateur Amazon RDS.

Pour créer l'instance de base de données RDS

1. Ouvrez la console.

   Ouvrez la console Amazon RDS et définissez la région sur USA West (Oregon). Dans le panneau de navigation, choisissez Tableau de bord RDS, puis Lancement d'une instance DB.
2. Spécifiez le moteur de base de données.

   Choisissez MySQL Community Edition comme moteur de base de données.
3. Refusez le déploiement multi-AZ.

   Choisissez No, this instance... (Non, cette instance…), puis Suivant. Vous n'avez pas besoin d'un déploiement multi-AZ pour cet exemple.
4. Configurez les paramètres de base.

   Sur la page DB Instance Details (Détails de l'instance de base de données), indiquez les valeurs suivantes :

   - Classe d'instance DB : db.t2.micro
   - Déploiement multi-AZ : Non
   - Stockage alloué : **5** Go
   - Identifiant de l'instance DB : **`nodeexample`**
   - Identifiant principal : **`opsworksuser`**
   - Mot de passe principal : mot de passe de votre choix

Enregistrez l'identifiant de l'instance, le nom d'utilisateur et le mot de passe en vue d'une utilisation ultérieure, acceptez les paramètres par défaut pour les autres options, puis choisissez Suivant.

5. Configurez les paramètres avancés.

   Sur la page Configuration de paramètres avancés, spécifiez les valeurs suivantes :

   - Nom de base de données : **`nodeexampledb`**
   - Groupe (s) de sécurité de base de données : AWS- OpsWorks -DB-Master-Server

   > **Note**
   >
   > Le groupe de sécurité AWS- OpsWorks -DB-Master-Server autorise uniquement les instances de votre stack à accéder à la base de données. Si vous souhaitez accéder directement à la base de données, attachez un groupe de sécurité supplémentaire à l'instance de base de données RDS avec les règles de trafic entrant appropriées. Pour plus d'informations, consultez Groupes de sécurité Amazon RDS. Vous pouvez également contrôler l'accès en plaçant l'instance dans un VPC. Pour de plus amples informations, veuillez consulter Running a Stack in a VPC.

   Enregistrez le nom de base de données en vue d'une utilisation ultérieure, acceptez les valeurs par défaut des autres paramètres et choisissez Lancement d'une instance DB.

La procédure suivante décrit comment créer un équilibreur de charge Elastic Load Balancing pour cet exemple. Pour plus d'informations, consultez le Guide de l'utilisateur Elastic Load Balancing.

Pour créer l'équilibreur de charge

1. Ouvrez la EC2 console Amazon.

   Ouvrez la EC2 console Amazon et assurez-vous que la région est définie sur USA Ouest (Oregon). Dans le panneau de navigation, choisissez Équilibreurs de charge, puis Créer un équilibreur de charge.

2. Définissez l'équilibreur de charge.

   Sur la page Définir un équilibreur de charge, spécifiez les paramètres suivants.

   - Nom – **`Node-LB`**
   - Create LB Inside — Mon VPC par défaut

   Acceptez les valeurs par défaut pour les autres options, puis choisissez Suivant.

3. Attribuez les groupes de sécurité.

   Sur la page Attribuer les groupes de sécurité, spécifiez les groupes suivants :

   - groupe de sécurité VPC par défaut
   - Serveur d'applications AWS- OpsWorks -NodeJS

   Choisissez Suivant. Sur la page Configurer les paramètres de sécurité, choisissez Suivant. Vous n'avez pas besoin d'un écouteur sécurisé pour cet exemple.

4. Configurez la vérification du statut.

   Sur la page Configurer la vérification de l'état, définissez Chemin de ping sur **`/`** et acceptez les valeurs par défaut pour les autres paramètres. Choisissez Suivant. Sur la page Ajouter EC2 des instances, choisissez Next. Sur la page Ajouter des balises, choisissez Réviser et créer. AWS OpsWorks Stacks se charge d'ajouter des EC2 instances à l'équilibreur de charge, et vous n'aurez pas besoin de balises pour cet exemple.

5. Créez l'équilibreur de charge.

   Sur la page Vérification, choisissez Créer pour créer l'équilibreur de charge.

Création de la pile

Vous avez maintenant tous les composants nécessaires pour créer la pile.

Pour créer la pile

1. Connectez-vous à la console AWS OpsWorks Stacks.

   Connectez-vous à la console AWS OpsWorks Stacks, puis choisissez Add Stack (Ajouter une pile).

2. Créez la pile.

   Pour créer une pile, choisissez Chef 11 stack (Pile Chef 11), puis spécifiez les paramètres suivants.

   - – **`NodeStack`**
   - Région — Ouest des États-Unis (Oregon)

     Vous pouvez créer une pile dans n'importe quelle région AWS, mais nous recommandons USA West (Oregon) pour les didacticiels.

   Choisissez Add Stack (Ajouter une pile). Pour plus d'informations sur les paramètres de configuration de pile, consultez Créer une pile.

3. Ajoutez une couche de serveur d'applications Node.js avec un équilibreur de charge attaché.

   Sur la NodeStackpage, choisissez Ajouter une couche, puis définissez les paramètres suivants :

   - Type de couche — Node.js App Server
   - Elastic Load Balancer — Node-LB

   Acceptez les valeurs par défaut pour les autres paramètres, puis choisissez Add Layer (Ajouter une couche).

4. Ajoutez les instances à la couche et démarrez-les.

   Dans le panneau de navigation, sélectionnez Instances, puis ajoutez deux instances à la couche Rails App Server, comme suit.

   1. Sous Node.js App Server, choisissez Ajouter une instance.

      Définissez Size (Taille) sur t2.micro, acceptez les valeurs par défaut pour les autres paramètres, puis choisissez Add Instance (Ajouter une instance).

   2. Choisissez +Instance, puis ajoutez une deuxième instance t2.micro à la couche dans un autre sous-réseau.

      Cela place l'instance dans une autre zone de disponibilité.

   3. Choisissez Add instance (Ajouter une instance).

4. Pour démarrer les deux instances, sélectionnez Start All Instances (Démarrer toutes les instances).

Vous avez attribué un équilibreur de charge Elastic Load Balancing à cette couche. Lorsqu'une instance entre ou quitte l'état en ligne, AWS OpsWorks Stacks enregistre ou désenregistre automatiquement l'instance auprès de l'équilibreur de charge.

> **Note**
>
> Pour une pile de production, nous vous recommandons de répartir vos instances de serveur d'applications sur plusieurs instances AZs. Si les utilisateurs ne peuvent pas se connecter à une zone de disponibilité, l'équilibreur de charge achemine le trafic entrant vers les instances des zones restantes, et votre site continue à fonctionner.

5. Inscrivez l'instance de base de données RDS auprès de la pile.

Dans le panneau de navigation, sélectionnez Resources (Ressources) et enregistrez l'instance de base de données RDS auprès de la pile, comme suit.

1. Choisissez l'onglet RDS, puis choisissez Show Unregistered RDS DB instances (Afficher les instances DB RDS non enregistrées).
2. Choisissez l'instance nodeexampledb, puis spécifiez les paramètres suivants :
   - Utilisateur : nom d'utilisateur principal que vous avez spécifié lors de la création de l'instance, dans cet exemple. **`opsworksuser`**.
   - Mot de passe : mot de passe principal que vous avez spécifié lors de la création de l'instance.
3. Choisissez Register with Stack pour ajouter l'instance de base de données RDS à la pile en tant que couche de service Amazon RDS.

> **Warning**
>
> AWS OpsWorks Stacks ne valide pas les valeurs d'utilisateur ou de mot de passe, il les transmet simplement à l'application. Si vous ne les entrez pas correctement, votre application ne peut pas se connecter à la base de données.

Pour ajouter l'instance de base de données RDS à la pile en tant que couche de service Amazon RDS, choisissez Register with Stack.

Déploiement de l'application

Vous devez stocker l'application dans un référentiel distant. Lorsque vous le déployez, AWS OpsWorks Stacks déploie le code et les fichiers associés du référentiel vers les instances du serveur d'applications. Pour plus de commodité, cet exemple utilise une archive publique Amazon Simple Storage Service (Amazon S3) comme référentiel, mais vous pouvez également utiliser plusieurs autres types de référentiels, notamment Git et Subversion. Pour de plus amples informations, veuillez consulter Source de l'application.

Pour déployer l'application

1. Empaquetez l'application dans un fichier d'archives.

   Créez une archive `.zip` du répertoire `nodedb` et des sous-répertoire nommée nodedb.zip. Vous pouvez également utiliser d'autres types de fichiers d'archives, y compris gzip, bzip2 et tarball. Notez que AWS OpsWorks Stacks ne prend pas en charge les archives non compressées. Pour de plus amples informations, veuillez consulter Source de l'application.

2. Téléchargez le fichier d'archive sur Amazon S3.

   `nodedb.zip`Téléchargez-le dans un compartiment Amazon S3, rendez le fichier public et copiez l'URL du fichier pour une utilisation ultérieure. Pour plus d'informations sur la création de compartiments et le chargement des fichiers, consultez Mise en route avec Amazon Simple Storage Service.

   > **Note**
   >
   > AWS OpsWorks Stacks peut également déployer des fichiers privés à partir d'un compartiment Amazon S3, mais pour des raisons de simplicité, cet exemple utilise un fichier public. Pour de plus amples informations, veuillez consulter Source de l'application.

3. Créez une application AWS OpsWorks Stacks.

   Revenez à la console AWS OpsWorks Stacks, dans le volet de navigation, choisissez Apps, puis choisissez Ajouter une application. Spécifiez les paramètres suivants :

   - Nom : `NodeDB`.

     Cette chaîne est le nom complet de l'application. Dans la plupart des cas, vous avez besoin du nom abrégé de l'application, que AWS OpsWorks Stacks génère à partir du nom d'affichage en transformant tous les caractères en minuscules et en supprimant la ponctuation. Dans cet exemple, le nom court est `nodedb`. Pour vérifier le nom court d'une application, après la création de l'application, choisissez l'application sur la page Apps (Applications) afin d'afficher sa page de détails.
   - Tapez —`Node.js`.
   - Type de source de données —`RDS`.
   - Instance de base de données — Choisissez l'instance de base de données Amazon RDS que vous avez enregistrée précédemment.
   - Nom de base de données — Spécifiez le nom de base de données que vous avez créé précédemment, `nodeexampledb` pour cet exemple.
   - Type de référentiel —`Http Archive`.

     Vous devez utiliser ce type de référentiel pour les fichiers Amazon S3 publics. Le type `S3 Archive` est utilisé uniquement pour les archives privées.
   - URL du référentiel : URL Amazon S3 du fichier d'archive.

   Utilisez les valeurs par défaut pour les paramètres restants, puis cliquez sur Add App (Ajouter une application) pour créer l'application.

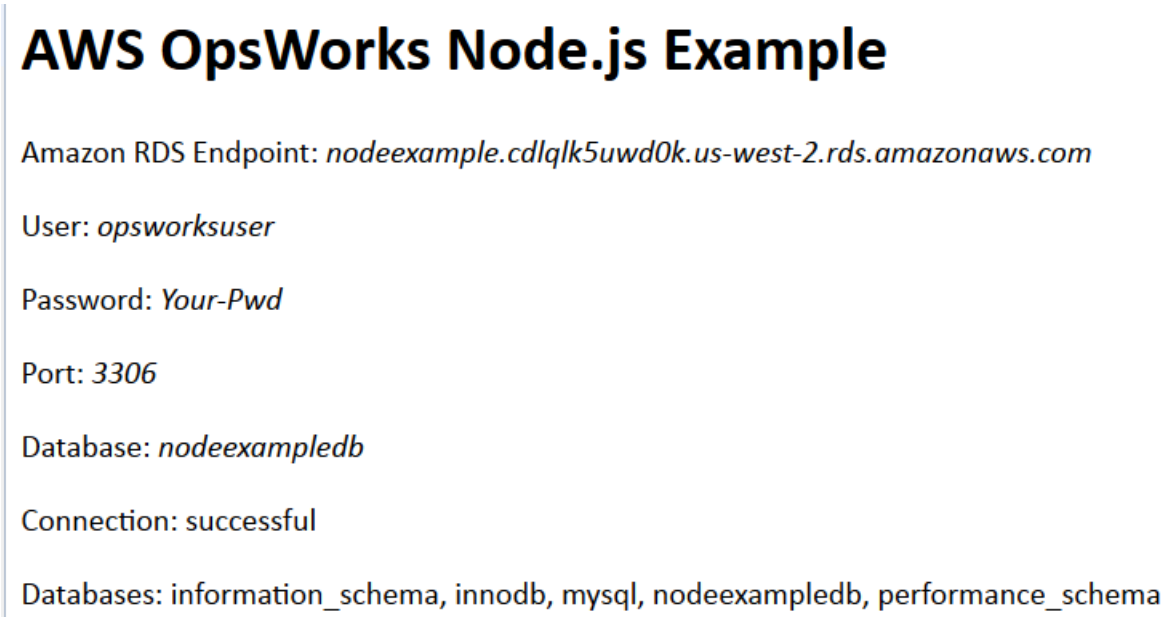
4. Déployez l'application.

   Accédez à la page Apps (Applications) et, dans la colonne Actions de l'application NodeDB, choisissez deploy (déployer). Choisissez ensuite Deploy pour déployer l'application sur les instances du serveur. AWS OpsWorks Stacks exécute les recettes de déploiement sur chaque instance, qui télécharge l'application depuis le référentiel et redémarre le serveur. Lorsque chaque instance est marquée d'une coche verte et que Status (Statut) a pour valeur successful (succès), le déploiement est terminé et l'application est prête à démarrer le traitement des demandes.

> **Note**
>
> Si le déploiement échoue, choisissez show (afficher) dans la colonne Log (Journal) pour afficher le journal Chef du déploiement. Les informations d'erreur se trouvent vers le bas.

5. Ouvrez l'application .

   Pour ouvrir l'application, choisissez successivement Layers (Couches), l'équilibreur de charge et le nom DNS de l'équilibreur de charge, ce qui envoie une requête HTTP à l'équilibreur de charge. Vous devriez voir quelque chose comme suit.

> **Note**
>
> AWS OpsWorks Stacks déploie automatiquement les applications sur les nouvelles instances lors de la configuration. Le déploiement manuel est obligatoire uniquement pour les instances en ligne. Pour de plus amples informations, veuillez consulter Déploiement d'applications. Pour une discussion générale du déploiement, y compris certaines stratégies de déploiement plus sophistiquées, consultez Gestion et déploiement des applications et livres de recettes.

Et maintenant ?

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette procédure pas à pas vous a guidé à travers les bases de la mise en place d'une pile de serveur d'applications Node.js simple. Voici quelques suggestions sur ce qu'il convient de faire ensuite.

Examiner le Node.js intégré dans les livres de recettes

Si vous souhaitez savoir comment les instances sont configurées en détail, consultez le livre de recettes intégré à la couche, opsworks_nodejs, qui contient les recettes et les fichiers associés que AWS OpsWorks Stacks utilise pour installer et configurer le logiciel, et le livre de recettes de déploiement intégré, qui contient les recettes utilisées par Stacks pour déployer les applications. AWS OpsWorks

Personnaliser la configuration du serveur

La pile de l'exemple est assez basique. Pour une utilisation en production, vous voudrez probablement personnaliser la pile. Pour de plus amples informations, veuillez consulter Personnalisation des piles AWS OpsWorks.

Ajouter la prise en charge SSL

Vous pouvez activer le support SSL pour votre application et fournir à AWS OpsWorks Stacks les certificats appropriés lors de la création de l'application. AWS OpsWorks Stacks installe ensuite les certificats dans le répertoire approprié. Pour de plus amples informations, veuillez consulter Utilisation de SSL.

Ajouter la mise en cache des données en mémoire

Les sites de niveau production améliorent souvent les performances en mettant en cache les données dans un magasin clé-valeur en mémoire, tel que Redis ou Memcache. Vous pouvez utiliser l'un ou l'autre avec une pile AWS OpsWorks Stacks. Pour plus d'informations, consultez ElastiCache Redis et Memcached.

Utiliser une stratégie de déploiement plus sophistiquée

L'exemple utilise une stratégie de déploiement d'application simple, qui déploie la mise à jour sur chaque instance de façon simultanée. Cette approche est simple et rapide, mais il n'y a aucune marge d'erreur. Si le déploiement échoue ou que la mise à jour rencontre des problèmes, chaque instance de votre pile en production peut être affectée, perturbant ou désactivant potentiellement votre site jusqu'à ce que vous puissiez résoudre le problème. Pour plus d'informations sur les stratégies de déploiement, consultez Gestion et déploiement des applications et livres de recettes.

Étendre la couche du serveur d'applications Node.js

Vous pouvez étendre la couche de diverses manières. Par exemple, vous pouvez implémenter des recettes pour exécuter des scripts sur les instances ou implémenter les raccordements de déploiement Chef afin de personnaliser le déploiement d'applications. Pour de plus amples informations, veuillez consulter Extension d'une couche.

Définir les variables d'environnement

Vous pouvez transmettre les données à votre application en définissant les variables d'environnement de l'application associée. Lorsque vous déployez l'application, AWS OpsWorks Stacks exporte ces variables afin que vous puissiez y accéder depuis votre application. Pour de plus amples informations, veuillez consulter Utilisation des variables d'environnement .

Personnalisation des piles AWS OpsWorks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Les couches intégrées de Stacks fournissent des fonctionnalités standard suffisantes à de nombreuses fins. Cependant, vous pouvez rencontrer un ou plusieurs des problèmes suivants :

- La configuration standard d'une couche intégrée est adéquate, mais pas idéale ; vous souhaitez l'optimiser pour vos besoins.

Par exemple, vous souhaiterez peut-être ajuster la configuration du serveur Nginx d'une couche de serveur Web statique en spécifiant vos propres valeurs pour des paramètres tels que le nombre maximum de processus de travail ou la `keepalivetimeout` valeur.

- Il est bien d'avoir une fonctionnalité de couche intégrée, mais il est important de l'étendre en installant des packages supplémentaires ou en exécutant des scripts d'installation personnalisés.

  Par exemple, vous souhaiterez peut-être étendre une couche PHP App Server en installant également un serveur Redis.

- Vous avez des exigences qui ne sont pas gérées par les couches intégrées.

  Par exemple, AWS OpsWorks Stacks n'inclut pas de couches intégrées pour certains serveurs de base de données populaires. Vous pouvez créer une couche personnalisée qui installe ces serveurs sur les instances de la couche.

- Vous exécutez une pile Windows qui prend en charge uniquement les couches personnalisées.

AWS OpsWorks Stacks propose différentes méthodes pour personnaliser les couches afin de répondre à vos besoins spécifiques. Les exemples suivants sont listés afin d'accroître la complexité et la puissance :

> **Note**
>
> Certaines de ces approches fonctionnent uniquement pour les piles Linux. Consultez les rubriques suivantes pour plus de détails.

- Utilisez le JSON personnalisé pour remplacer les paramètres par défaut de AWS OpsWorks Stacks.
- Implémentez un livre de recettes Chef personnalisé avec un fichier d'attributs qui remplace les paramètres par défaut de AWS OpsWorks Stacks.
- Implémentez un livre de recettes Chef personnalisé avec un modèle qui remplace ou étend un modèle AWS OpsWorks Stacks par défaut.
- Implémentez un livre de recettes Chef personnalisé avec une recette simple qui exécute un script shell.

- Implémentez un livre de recettes Chef personnalisé avec des recettes qui effectuent des tâches telles que la création et la configuration des répertoires, l'installation de packages, la création de fichiers de configuration, le déploiement d'applications, etc.

Vous pouvez également remplacer les recettes, en fonction de la version de Chef de la pile et du système d'exploitation.

- Avec les piles Chef 0.9 et 11.4, vous ne pouvez pas remplacer une recette intégrée en implémentant une recette personnalisée avec les mêmes noms de livre de recettes et de recette.

  Pour chaque événement du cycle de vie, AWS OpsWorks Stacks exécute toujours les recettes intégrées en premier, suivies de toutes les recettes personnalisées. Etant donné que ces versions de Chef n'exécutent pas deux fois une recette avec les mêmes noms de livre de recettes et de recette, la recette intégrée est prioritaire et la recette personnalisée ne sera pas exécutée.
- Vous pouvez remplacer les recettes intégrées sur les piles de Chef 11.10.

  Pour de plus amples informations, veuillez consulter Installation et priorité des livres de recettes.
- Vous ne pouvez pas remplacer les recettes intégrées sur les piles Windows.

  La façon dont AWS OpsWorks Stacks gère les exécutions de Chef pour Windows Stacks ne permet pas de remplacer les recettes intégrées.

> **Note**
>
> Étant donné que de nombreuses techniques utilisent des livres de recettes personnalisés, vous devez d'abord les lire Livres de recettes et recettes si vous n'êtes pas déjà familiarisé avec l'implémentation des livres de recettes. Principes de base des livre de recettesfournit une introduction détaillée à la mise en œuvre de livres de recettes personnalisés et Implémentation de Cookbooks for Stacks AWS OpsWorks explique en détail comment implémenter des livres de recettes pour les instances AWS OpsWorks Stacks.

Rubriques

- Personnalisation de la configuration AWS OpsWorks des piles en remplaçant les attributs
- Extension des fichiers de configuration de AWS OpsWorks Stacks à l'aide de modèles personnalisés

- Extension d'une couche
- Création d'une couche serveur Tomcat personnalisée
- Attributs de déploiement et de configuration de pile

Personnalisation de la configuration AWS OpsWorks des piles en remplaçant les attributs

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Pour les stacks Windows et les stacks Chef 12 Linux, AWS OpsWorks Stacks utilise des exécutions Chef distinctes pour les recettes intégrées et les recettes personnalisées. Cela signifie que vous ne pouvez pas utiliser les techniques décrites dans cette section pour remplacer les attributs intégrés des piles Windows et Chef 12 Linux.

Les recettes et les modèles dépendent de divers attributs de Chef pour les informations propres à l'instance ou à la pile, par exemple les paramètres de configuration de couche ou de serveur d'applications. Ces attributs ont plusieurs sources :

- JSON personnalisé : vous pouvez éventuellement spécifier des attributs JSON personnalisés lorsque vous créez, mettez à jour ou clonez une pile, ou lorsque vous déployez une application.
- Attributs de configuration de pile —AWS OpsWorks Stacks définit ces attributs pour contenir les informations de configuration de la pile, y compris les informations que vous spécifiez dans les paramètres de la console.
- Attributs de déploiement : AWS OpsWorks définit les attributs liés au déploiement pour les événements de déploiement.
- Attributs du livre de recettes — Les livres de recettes intégrés et personnalisés incluent généralement un ou plusieurs fichiers d'attributs, qui contiennent des attributs représentant des

valeurs spécifiques au livre de recettes, telles que les paramètres de configuration du serveur d'applications.

- Chef —L'outil Ohai de Chef définit les attributs qui représentent une grande variété de paramètres de configuration du système, tels que le type de processeur et la mémoire installée.

Pour une liste complète des attributs de configuration et de déploiement de la pile, ainsi que des attributs intégrés des livres de recettes, consultez Attributs de déploiement et de configuration de pile : Linux et Attributs des livres de recettes intégrés. Pour plus d'informations sur les attributs Ohai, consultez Ohai.

Lorsqu'un événement du cycle de vie tel que Deploy ou Configure se produit, ou si vous exécutez une commande de pile comme `execute_recipes` ou `update_packages`, AWS OpsWorks Stacks effectue les opérations suivantes :

- Envoie une commande correspondante à l'agent sur chaque instance concernée.

  L'agent exécute les recettes appropriées. Par exemple, pour un événement Deploy, l'agent exécute les recettes Deploy intégrées, suivies des recettes Deploy personnalisées.
- Fusionne les attributs personnalisés JSON et de déploiement avec les attributs de configuration de la pile et les installe sur les instances.

Les attributs du JSON personnalisé, les attributs de configuration et de déploiement de la pile, les attributs des livres de recettes et les attributs Ohai sont fusionnés dans un objet de nœud, qui donne les valeurs d'attributs aux recettes. En règle générale, une instance n'a pas d'état en ce qui concerne les attributs de configuration de la pile, y compris n'importe quel JSON personnalisé. Lorsque vous exécutez une commande de déploiement ou de pile, les recettes associées utilisent les attributs de configuration de la pile qui ont été téléchargés avec la commande.

Rubriques

- Priorité des attributs
- Remplacement des attributs par un JSON personnalisé
- Remplacer les attributs des AWS OpsWorks piles à l'aide d'attributs de livre de recettes personnalisés

Priorité des attributs

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Si un attribut est défini de façon unique, Chef l'intègre simplement dans l'objet de nœud. Toutefois, n'importe quelle source d'attribut peut définir un attribut, il est donc possible pour le même attribut d'avoir plusieurs définitions avec des valeurs différentes. Par exemple, le livre de recettes intégré `apache2` définit `node[:apache][:keepalive]`, mais vous pouvez également définir cet attribut dans le JSON personnalisé ou dans un livre de recettes personnalisé. Si un attribut possède plusieurs définitions, elles sont évaluées dans un ordre qui est décrit plus loin et l'objet de nœud reçoit la définition avec la priorité la plus élevée.

Un attribut est défini comme suit :

```
node.type[:attribute][:sub_attribute][:...]=value
```

Si un attribut possède plusieurs définitions, le type détermine quelle définition est prioritaire, et cette définition est incorporée dans l'objet du nœud. AWS OpsWorks Stacks utilise les types d'attributs suivants :

- default —Il s'agit du type le plus courant, et cela signifie essentiellement « utiliser cette valeur si l'attribut n'a pas déjà été défini ». Si toutes les définitions d'un attribut sont de type `default`, la première définition dans l'ordre d'évaluation a priorité et les valeurs suivantes sont ignorées. Notez que AWS OpsWorks Stacks définit toutes les définitions d'attributs de configuration et de déploiement de la pile sur le `default` type.
- normal —Les attributs de ce type remplacent tous `default` `normal` les attributs définis précédemment dans l'ordre d'évaluation. Par exemple, si le premier attribut provient d'un livre de recettes intégré et possède un type `default` et si un deuxième attribut a été défini par l'utilisateur avec un type `normal`, la deuxième définition sera prioritaire.
- set —Il s'agit d'un type obsolète que vous pourriez voir dans les anciens livres de cuisine. Il a été remplacé par `normal`, qui a la même priorité.

Chef prend en charge plusieurs types d'attribut supplémentaires, dont un type `automatic` qui est prioritaire sur toutes les autres définitions d'attributs. Les définitions d'attributs générées par l'outil Ohai de Chef ont toutes le type `automatic` et sont donc en lecture seule. Ce n'est généralement pas un problème, car il n'y a aucune raison de les remplacer et ils sont distincts des attributs de AWS OpsWorks Stacks. Toutefois, veillez à nommer vos attributs personnalisés des livres de recettes différemment des attributs Ohai. Pour plus d'informations, consultez À propos des attributs.

> **Note**
>
> L'outil Ohai est un exécutable que vous pouvez exécuter à partir de la ligne de commande. Pour lister les attributs Ohai d'une instance, connectez-vous à l'instance et exécutez `ohai` dans une fenêtre de terminal. Sachez qu'elle produit une très longue sortie.

Voici les étapes qui intègrent les différentes définitions d'attributs à l'objet de nœud :

1. Fusionnez les attributs personnalisés de configuration avec les attributs de configuration et de déploiement de la pile.

   Les attributs JSON personnalisés peuvent être définis pour la pile ou pour un déploiement particulier. Ils sont les premiers dans l'ordre d'évaluation et ont le type `normal`. Si un ou plusieurs attributs de configuration de la pile sont aussi définis dans le JSON personnalisé, les valeurs du JSON personnalisé ont priorité. Sinon, AWS OpsWorks Stacks intègre simplement les attributs personnalisés de JSON dans la configuration de la pile.
2. Fusionnez les attributs de déploiement personnalisés JSON avec les attributs de déploiement et de configuration de la pile.

   Les attributs de déploiement personnalisés JSON ont le type `normal`, ce qui signifie qu'ils ont la priorité sur le JSON de configuration de pile intégré et personnalisé et sur le JSON de déploiement intégré.
3. Fusionnez les attributs de configuration et de déploiement de pile dans l'objet de nœud de l'instance.
4. Fusionnez les attributs des livres de recettes intégrés de l'instance dans l'objet de nœud.

   Les attributs intégrés des livres de recettes ont tous le type `default`. Si le ou les attributs intégrés du livre de recettes sont également définis dans les attributs de configuration et de déploiement de la pile, généralement parce que vous les avez définis avec du code JSON personnalisé, les définitions de configuration de la pile ont priorité sur les définitions du livre de recettes intégrées.

Tous les autres attributs intégrés des livres de recettes sont simplement intégrés dans l'objet de nœud.

5. Fusionnez les attributs des livres de recettes personnalisés de l'instance dans l'objet de nœud.

   Les attributs personnalisés des livres de recettes ont généralement le type `normal` ou `default`. Des attributs uniques sont intégrés dans l'objet de nœud. Si des attributs de livre de recettes personnalisés sont également définis aux étapes 1 à 3 (généralement parce que vous les avez définis avec du JSON personnalisé), la priorité dépend du type de l'attribut de livre de recettes personnalisé :

   - Les attributs définis aux étapes 1 à 3 ont priorité sur les attributs personnalisés du livre de recettes`default`.
   - Les `normal` attributs personnalisés du livre de cuisine ont priorité sur les définitions des étapes 1 à 3.

> **Important**
>
> N'utilisez pas les attributs personnalisés `default` des livres de recettes pour remplacer les attributs intégrés des livres de recettes ou les attributs de configuration de la pile. Dans la mesure où les attributs personnalisés des livres de recettes sont évalués en dernier, les attributs `default` ont la priorité la plus basse et ne peuvent rien remplacer.

Remplacement des attributs par un JSON personnalisé

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Comme AWS OpsWorks Stacks gère les exécutions de Chef différemment pour les piles Windows et pour les piles Linux, vous ne pouvez pas utiliser les techniques décrites dans cette section pour les piles Windows.

Le moyen le plus simple de remplacer un attribut AWS OpsWorks Stacks consiste à le définir dans du JSON personnalisé, qui a priorité sur les attributs de configuration et de déploiement de la pile, ainsi que sur les attributs de livre de recettes intégrés et personnalisés. `default` Pour de plus amples informations, veuillez consulter Priorité des attributs.

> **Important**
>
> Vous devez remplacer les attributs de configuration et de déploiement de la pile avec soin. Par exemple, le remplacement des attributs dans l'espace de noms `opsworks` peut interférer avec les recettes intégrées. Pour de plus amples informations, veuillez consulter Attributs de déploiement et de configuration de pile.

Vous pouvez également utiliser le JSON personnalisé pour définir les attributs uniques, généralement pour transmettre des données à vos recettes personnalisées. Les attributs sont simplement intégrés à l'objet de nœud et les recettes peuvent les référence à l'aide de la syntaxe de nœud standard de Chef.

Procédure pour spécifier le JSON personnalisé

Pour utiliser le JSON personnalisé afin de remplacer une valeur d'attribut, vous devez tout d'abord déterminer le nom qualifié de l'attribut. Vous créez ensuite un objet JSON qui contient les attributs que vous voulez remplacer, défini sur vos valeurs préférées. Pour plus de commodité, les documents Attributs de déploiement et de configuration de pile : Linux et Attributs des livres de recettes intégrés utilisent généralement les attributs de configuration de la pile, de déploiement et de livres de recettes intégrés, avec leurs noms pleinement qualifiés.

Les relations parent-enfant de l'objet doivent correspondre aux nœuds pleinement qualifiés appropriés de Chef. Par exemple, supposons que vous souhaitiez modifier les attributs suivants d'Apache :

- L'attribut `keepalivetimeout`, dont le nœud est `node[:apache][:keepalivetimeout]` et a une valeur par défaut de `3`.
- L'attribut `logrotate` `schedule`, dont le nœud est `node[:apache][:logrotate][:schedule]` et a une valeur par défaut de `"daily"`.

Pour remplacer les attributs et définir les valeurs sur `5` et `"weekly"`, respectivement, vous devez utiliser le JSON personnalisé suivant :

```
{
  "apache" : {
    "keepalivetimeout" : 5,
    "logrotate" : {
       "schedule" : "weekly"
    }
  }
}
```

Quand spécifier le JSON personnalisé

Vous pouvez spécifier une structure de JSON personnalisé pour les tâches suivantes :

- Créer une pile
- Mettre à jour une pile
- Exécuter une commande de pile
- Cloner une pile
- Déployer une application

Pour chaque tâche, AWS OpsWorks Stacks fusionne les attributs JSON personnalisés avec les attributs de configuration et de déploiement de la pile et les envoie aux instances pour qu'elles soient fusionnées dans l'objet du nœud. Toutefois, notez les points suivants :

- Si vous spécifiez le JSON personnalisé lorsque vous créez, clonez ou mettez à jour une pile, les attributs sont fusionnés avec les attributs de configuration et de déploiement de la pile pour tous les événements du cycle de vie et les commandes de pile qui suivront.
- Si vous spécifiez un JSON personnalisé pour un déploiement, les attributs sont fusionnés dans les attributs de configuration et de déploiement de la pile uniquement pour l'événement correspondant.

Si vous souhaitez utiliser ces attributs personnalisés pour les déploiements suivants, vous devez spécifier explicitement le JSON personnalisé à nouveau.

Il est important de se souvenir que les attributs affectent uniquement l'instance lorsqu'ils sont utilisés par des recettes. Si vous remplacez une valeur d'attribut, mais qu'aucune recette ultérieure ne fait référence à l'attribut, la modification n'a aucun effet. Vous devez soit vous assurer que le JSON personnalisé est envoyé avant l'exécution des recettes associées, soit veiller à ce que les recettes appropriées soient exécutées à nouveau.

Bonnes pratiques pour le JSON personnalisé

Vous pouvez utiliser du JSON personnalisé pour remplacer n'importe quel attribut AWS OpsWorks Stacks, mais la saisie manuelle des informations est quelque peu fastidieuse et ne fait l'objet d'aucun contrôle de source. Le JSON personnalisé est plus adapté pour les raisons suivantes :

- Lorsque vous voulez remplacer uniquement un petit nombre d'attributs et vous n'avez pas besoin d'utiliser les livres personnalisés dans d'autres buts.

  Avec le JSON personnalisé, vous pouvez éviter la complexité de la configuration et de la gestion d'un référentiel de livres de recettes uniquement pour remplacer deux attributs.
- Valeurs sensibles, telles que les mots de passe ou les clés d'authentification.

  Les attributs des livres de recettes sont stockés dans un référentiel, c'est pourquoi les informations sensibles risquent d'être compromises. C'est pourquoi il est préférable de définir les attributs avec des valeurs fictives et d'utiliser le JSON personnalisé pour définir les valeurs réelles.
- Valeurs qui sont censées varier.

  Par exemple, une pratique recommandée consiste à avoir votre pile de production prise en charge par des piles de développement et intermédiaires distinctes. Supposons que ces piles prennent en charge une application qui accepte les paiements. Si vous utilisez le JSON personnalisé pour spécifier le point de terminaison du paiement, vous pouvez spécifier une URL de test pour votre pile intermédiaire. Lorsque vous êtes prêt à migrer une pile mise à jour vers votre pile de production, vous pouvez utiliser les mêmes livres de recettes et le JSON personnalisé pour définir le point de terminaison de paiement sur l'URL de production.
- Les valeurs qui sont propres à une commande spécifique de pile ou de déploiement.

Remplacer les attributs des AWS OpsWorks piles à l'aide d'attributs de livre de recettes personnalisés

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Pour les piles Windows, AWS OpsWorks Stacks utilise des exécutions Chef distinctes pour les recettes intégrées et les recettes personnalisées. Cela signifie que vous ne pouvez pas utiliser les techniques décrites dans cette section pour remplacer les attributs intégrés pour les piles Windows.

Le JSON personnalisé est un moyen pratique de remplacer la configuration de la pile AWS OpsWorks Stacks et les attributs intégrés du livre de recettes, mais il présente certaines limites. Vous devez plus particulièrement entrer le JSON personnalisé manuellement chaque utilisation, ce qui signifie que vous n'avez aucun moyen solide pour gérer les définitions. Une meilleure approche consiste souvent à utiliser des fichiers d'attributs de livres de recettes personnalisés pour remplacer les attributs intégrés. Cela vous permet de placer les définitions sous le contrôle de code source.

La procédure d'utilisation de fichiers d'attributs personnalisés pour remplacer les définitions de AWS OpsWorks Stacks est simple.

Pour remplacer les définitions d' AWS OpsWorks attributs Stacks

1. Définissez un référentiel de livres de recettes, comme décrit dans Livres de recettes et recettes.
2. Créez un livre de recettes avec le même nom que le livre de recettes intégré qui contient les attributs que vous voulez remplacer. Par exemple, pour remplacer les attributs Apache, le livre de recettes doit être nommé apache2.
3. Ajoutez un dossier `attributes` au livre de recettes et ajoutez un fichier dans ce dossier nommé `customize.rb`.

4. Ajoutez une définition d'attribut dans le fichier pour chacun des attributs du livre de recettes intégré que vous voulez remplacer, avec votre valeur préférée. L'attribut doit être de `normal` type ou supérieur et porter exactement le même nom de nœud que l'attribut AWS OpsWorks Stacks correspondant. Pour une liste détaillée des attributs de AWS OpsWorks Stacks, y compris les noms des nœuds, reportez-vous aux sections Attributs de déploiement et de configuration de pile : Linux etAttributs des livres de recettes intégrés. Pour plus d'informations sur les attributs et les fichiers d'attributs, consultez À propos des fichiers d'attributs.

   > **Important**
   >
   > Vos attributs doivent être `normal` de type pour remplacer les attributs de AWS OpsWorks Stacks ; les `default` types n'ont pas de priorité. Par exemple, si votre fichier `customize.rb` contient une définition d'attribut `default[:apache][:keepalivetimeout] = 5`, l'attribut correspondant du fichier d'attributs intégré `apache.rb` est évalué en premier et il est prioritaire. Pour de plus amples informations, veuillez consulter Remplacement des attributs.

5. Répétez les étapes 2 à 4 pour chaque livre de recettes intégré dont les attributs doivent être remplacés.
6. Activez des livres de recettes personnalisés pour votre pile et fournissez les informations requises pour que AWS OpsWorks Stacks télécharge vos livres de recettes sur les instances de la pile. Pour de plus amples informations, veuillez consulter Installation de livres de recettes personnalisés.

> **Note**
>
> Pour une présentation complète de cette procédure, consultez Remplacement des attributs intégrés.

L'objet de nœud utilisé par les événements du cycle de vie, les commandes de déploiement et les commandes de pile ultérieurs contiendra désormais vos définitions d'attributs au lieu des valeurs AWS OpsWorks Stacks.

Par exemple, pour remplacer les paramètres intégrés `keepalivetimeout` et `logrotate schedule` dans Procédure pour spécifier le JSON personnalisé, ajoutez un livre de recettes

`apache2`apache à votre référentiel et ajoutez un fichier `customize.rb` au dossier `attributes` du livre de recettes avec le contenu suivant.

```
normal[:apache][:keepalivetimeout] = 5
normal[:apache][:logrotate][:schedule] = 'weekly'
```

> **Important**
>
> Vous ne devez pas remplacer les attributs AWS OpsWorks Stacks en modifiant une copie du fichier d'attributs intégré associé. Si, par exemple, vous copiez `apache.rb` dans votre dossier `apache2/attributes` et que vous modifiez certains de ses paramètres, vous remplacez essentiellement tous les attributs du fichier intégré. Les recettes utiliseront les définitions d'attribut de votre copie et ignoreront le fichier intégré. Si AWS OpsWorks Stacks modifie ultérieurement le fichier d'attributs intégré, les recettes n'auront pas accès aux changements, sauf si vous mettez à jour manuellement votre copie.
> Pour éviter cette situation, tous les livres de recettes intégrés comportent un fichier d'attributs `customize.rb` vide, qui est nécessaire dans tous les modules via une directive `include_attribute`. En remplaçant les attributs de votre copie du fichier `customize.rb`, vous affectez uniquement ces attributs. Les recettes obtiendront toutes les autres valeurs d'attribut des fichiers d'attributs intégrés et obtiendront automatiquement les valeurs actuelles de tous les attributs que vous n'aurez pas remplacés.
> Cette approche vous permet de limiter le nombre d'attributs dans votre référentiel de livres de recettes, ce qui réduit votre charge de maintenance et rend les mises à niveau futures plus faciles à gérer.

Extension des fichiers de configuration de AWS OpsWorks Stacks à l'aide de modèles personnalisés

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Comme AWS OpsWorks Stacks gère les exécutions de Chef différemment pour les piles Windows et pour les piles Linux, vous ne pouvez pas utiliser les techniques décrites dans cette section pour les piles Windows.

AWS OpsWorks Stacks utilise des modèles pour créer des fichiers tels que des fichiers de configuration, qui dépendent généralement des attributs de nombreux paramètres. Si vous utilisez du JSON personnalisé ou des attributs de livre de recettes personnalisés pour remplacer les définitions de AWS OpsWorks Stacks, vos paramètres préférés sont incorporés dans les fichiers de configuration à la place des paramètres Stacks. AWS OpsWorks Cependant, AWS OpsWorks Stacks ne spécifie pas nécessairement un attribut pour chaque paramètre de configuration possible ; il accepte les valeurs par défaut de certains paramètres et en code dur d'autres directement dans le modèle. Vous ne pouvez pas utiliser du JSON personnalisé ou des attributs de livre de recettes personnalisés pour spécifier les paramètres préférés s'il n'existe aucun attribut AWS OpsWorks Stacks correspondant.

Vous pouvez étendre le fichier de configuration de façon à inclure des paramètres de configuration supplémentaires en créant un modèle personnalisé. Vous pouvez ensuite ajouter des paramètres de configuration ou tout autre contenu dont vous avez besoin au fichier et remplacer les paramètres codés en dur. Pour plus d'informations sur les modèles, consultez Modèles.

> **Note**
>
> Vous pouvez remplacer n'importe quel modèle intégré, sauf opsworks-agent.monitrc.erb.

Pour créer un modèle personnalisé

1. Créez un livre de recettes avec les mêmes noms de structure et de répertoire que le livre de recettes intégré. Ensuite, créez un modèle de fichier dans le répertoire approprié avec le même nom que celui du modèle intégré que vous souhaitez personnaliser. Par exemple, pour utiliser un modèle personnalisé afin d'étendre le fichier de configuration `httpd.conf` Apache, vous devez implémenter un livre de recettes `apache2` dans votre référentiel et votre modèle de fichier doit être `apache2/templates/default/apache.conf.erb`. L'utilisation des mêmes noms permet à AWS OpsWorks Stacks de reconnaître le modèle personnalisé et de l'utiliser à la place du modèle intégré.

L'approche la plus simple consiste à simplement copier le fichier modèle intégré du GitHubréférentiel du livre de recettes intégré dans votre livre de recettes et à le modifier selon vos besoins.

> **Important**
>
> Ne copiez aucun fichier du livre de recettes intégré à l'exception des fichiers de modèles que vous souhaitez personnaliser. Des copies d'autres types de fichiers de livres de recettes, par exemple des recettes, créent des ressources de Chef en double et peuvent entraîner des erreurs.

Le livre de recettes peut également inclure des attributs personnalisés, des recettes et des fichiers associés, mais leurs noms de fichiers ne doivent pas dupliquer les noms de fichiers intégrés.

2. Personnalisez le modèle de fichier pour générer un fichier de configuration correspondant à vos besoins. Vous pouvez ajouter des paramètres, supprimer des paramètres existants, remplacer des attributs codés en dur, etc.
3. Si vous ne l'avez pas déjà fait, modifiez les paramètres de la pile afin d'activer les livres de recettes personnalisés et de spécifier votre référentiel de livres de recettes. Pour de plus amples informations, veuillez consulter Installation de livres de recettes personnalisés.

> **Note**
>
> Pour une présentation complète de cette procédure, consultez Remplacement des modèles intégrés.

Il n'est pas nécessaire d'implémenter des recettes ou d'ajouter des recettes à la configuration des couches pour remplacer un modèle. AWS OpsWorks Stacks exécute toujours les recettes intégrées. Lorsqu'il exécute la recette qui crée le fichier de configuration, il utilise automatiquement votre modèle personnalisé au lieu du modèle intégré.

> **Note**
>
> Si AWS OpsWorks Stacks apporte des modifications au modèle intégré, votre modèle personnalisé risque de se désynchroniser et de ne plus fonctionner correctement. Supposons, par exemple, que votre modèle fasse référence à un fichier dépendant et que le nom du fichier change. AWS OpsWorks Stacks n'apporte pas souvent de telles modifications, et lorsqu'un modèle change, il répertorie les modifications et vous donne la possibilité de passer à une nouvelle version. Vous devez surveiller les modifications apportées au référentiel AWS OpsWorks Stacks et mettre à jour manuellement votre modèle si nécessaire.

Extension d'une couche

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Parfois, vous avez besoin de personnaliser une couche intégrée au-delà de ce qui peut être géré en modifiant les attributs AWS OpsWorks Stacks ou en personnalisant les modèles. Par exemple, supposons que vous ayez besoin de créer des liens symboliques, de définir des modes de fichier ou de dossier, d'installer des packages supplémentaires, etc. Vous devez étendre les couches personnalisées pour offrir plus que les fonctionnalités minimales. Dans ce cas, vous devrez mettre en place un ou plusieurs livres de recettes personnalisés, avec des recettes pour gérer les tâches de personnalisation. Cette rubrique fournit quelques exemples de la façon d'utiliser les recettes pour étendre une couche.

Si vous utilisez Chef pour la première fois, lisez Les bases des livres de recettes ; ce didacticiel présente les notions de base qui expliquent comment implémenter les livres de recettes et exécuter une grande variété de tâches courantes. Pour obtenir un exemple détaillé de la façon d'implémenter une couche personnalisée, consultez Création d'une couche serveur Tomcat personnalisée.

Rubriques

- Utilisation de recettes pour exécuter des scripts

- Utilisation des raccordements de déploiement Chef
- Exécution de tâches cron sur les instances Linux
- Installation et configuration des packages sur les instances Linux

Utilisation de recettes pour exécuter des scripts

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Si vous avez déjà un script qui exécute les tâches de personnalisation nécessaires, l'approche la plus simple pour l'extension d'une couche consiste souvent à implémenter une recette simple pour exécuter le script. Vous pouvez ensuite assigner la recette à des événements de cycle de vie appropriés, généralement Setup ou Deploy, ou utiliser la commande de pile `execute_recipes` pour exécuter la recette manuellement.

L'exemple suivant exécute un script shell sur des instances Linux, mais vous pouvez utiliser la même approche pour d'autres types de scripts, y compris les PowerShell scripts Windows.

```
cookbook_file "/tmp/lib-installer.sh" do
  source "lib-installer.sh"
  mode 0755
end

execute "install my lib" do
  command "sh /tmp/lib-installer.sh"
end
```

La ressource `cookbook_file` représente un fichier qui est stocké dans un sous-répertoire du répertoire `files` d'un livre de recettes et transfère le fichier vers un emplacement spécifié sur l'instance. Cet exemple transfère un script shell, `lib-installer.sh`, vers le répertoire `/tmp` de l'instance et définit le mode du fichier sur `0755`. Pour plus d'informations, consultez cookbook_file.

La ressource `execute` représente une commande, telle qu'une commande shell. Cet exemple exécute `lib-installer.sh`. Pour plus d'informations, consultez execute.

Vous pouvez également exécuter un script en l'intégrant à une recette. L'exemple suivant exécute un script bash, mais Chef prend également en charge Csh, Perl, Python et Ruby.

```
script "install_something" do
  interpreter "bash"
  user "root"
  cwd "/tmp"
  code <<-EOH
    #insert bash script
  EOH
end
```

La ressource `script` représente un script. L'exemple spécifie un interpréteur bash, définit l'utilisateur sur `"root"`et le répertoire de travail sur `/tmp`. Ensuite, il exécute le script bash du bloc `code`, lequel peut inclure autant de lignes que requis. Pour plus d'informations, consultez script.

Pour plus d'informations sur l'utilisation de recettes pour exécuter des scripts, consultez Exemple 7 : Exécution des commandes et des scripts. Pour un exemple d'exécution d'un PowerShell script sur une instance Windows, consultezExécution d'un PowerShell script Windows.

Utilisation des raccordements de déploiement Chef

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez personnaliser le déploiement en implémentant une recette personnalisée pour exécuter les tâches requises et en l'affectant à l'événement Deploy de la couche appropriée. Une approche alternative et parfois plus simple, en particulier si vous n'avez pas besoin d'implémenter un livre de recettes à d'autres fins, consiste à utiliser les hooks de déploiement Chef pour exécuter votre code de personnalisation. En outre, les recettes Deploy personnalisées s'exécutent une fois que le

déploiement a déjà été réalisé par les recettes intégrées. Les raccordements de déploiement vous permettent d'interagir lors d'un déploiement : par exemple, une fois que le code de l'application a été extrait du référentiel, mais avant qu'Apache n'ait redémarré.

Chef déploie les applications en quatre étapes :

- Checkout —Télécharge les fichiers depuis le référentiel
- Migrer —Exécute une migration, selon les besoins
- Symlink —Crée des liens symboliques
- Redémarrer : redémarre l'application

Les raccordements de déploiement Chef offrent un moyen simple de personnaliser un déploiement en exécutant le cas échéant une application Ruby fournie par l'utilisateur, une fois chaque étape terminée. Pour utiliser les raccordements de déploiement, implémentez une ou plusieurs applications Ruby et placez-les dans le répertoire `/deploy`. (Si votre application n'a pas de répertoire `/deploy`, créez-en un au niveau `APP_ROOT`.) L'application doit avoir l'un des noms suivants, qui détermine quand elle s'exécute.

- `before_migrate.rb` s'exécute une fois l'étape d'extraction terminée, mais avant la migration.
- `before_symlink.rb` s'exécute une fois l'étape d'extraction terminée, mais avant les liens symboliques.
- `before_restart.rb` s'exécute une fois l'étape des liens symboliques terminée, mais avant le redémarrage.
- `after_restart.rb` s'exécute une fois le redémarrage terminé.

Les raccordements de déploiement Chef peuvent accéder à l'objet nœud en utilisant la syntaxe nœud standard, tout comme les recettes. Les raccordements de déploiement peuvent également accéder aux valeurs des variables d'environnement d'application que vous avez spécifiées. Cependant, vous devez utiliser `new_resource.environment["VARIABLE_NAME"]` pour accéder à valeur de la variable au lieu de `ENV["VARIABLE_NAME"]`.

Exécution de tâches cron sur les instances Linux

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement

aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une tâche cron Linux demande au processus cron d'exécuter une ou plusieurs commandes selon un calendrier spécifié. Par exemple, supposons que votre pile prenne en charge une application de commerce en ligne PHP. Vous pouvez configurer une tâche cron pour que le serveur vous envoie un rapport des ventes à une heure définie chaque semaine. Pour plus d'informations sur cron, consultez cron sur Wikipedia. Pour plus d'informations sur la façon d'exécuter directement une tâche cron sur une instance ou un ordinateur Linux, consultez Présentation et utilisation de cron et crontab sur le site de la base de connaissances de l'université d'Indiana.

Même si vous pouvez configurer manuellement des tâches `cron` sur les instances Linux en vous connectant à elles avec SSH et en modifiant leurs entrées `crontab`, un avantage essentiel d' AWS OpsWorks Stacks est que vous pouvez lui ordonner d'exécuter la tâche sur une couche complète d'instances. La procédure suivante décrit comment configurer une `cron` tâche sur les instances d'une couche PHP App Server, mais vous pouvez utiliser la même approche avec n'importe quelle couche.

Pour configurer une tâche **`cron`** sur les instances d'une couche

1. Implémentez un livre de recettes avec une recette ayant une ressource `cron` qui configure la tâche. L'exemple suppose que la recette se nomme `cronjob.rb` ; les détails de l'implémentation sont décrits ultérieurement. Pour plus d'informations sur les livres de recettes et les recettes, consultez Livres de recettes et recettes.
2. Installez le livre de recettes sur votre pile. Pour de plus amples informations, veuillez consulter Installation de livres de recettes personnalisés.
3. Demandez à AWS OpsWorks Stacks d'exécuter automatiquement la recette sur les instances de la couche en l'affectant aux événements du cycle de vie suivants. Pour de plus amples informations, veuillez consulter Exécution automatique des recettes.

   - Configuration : l'attribution `cronjob.rb` à cet événement indique à AWS OpsWorks Stacks d'exécuter la recette sur toutes les nouvelles instances.
   - Déployer : l'attribution `cronjob.rb` à cet événement indique à AWS OpsWorks Stacks d'exécuter la recette sur toutes les instances en ligne lorsque vous déployez ou redéployez une application sur la couche.

Vous pouvez aussi exécuter manuellement la recette sur les instances en ligne en utilisant la commande de pile `Execute Recipes`. Pour de plus amples informations, veuillez consulter Exécution des commandes de pile.

L'exemple `cronjob.rb` suivant configure une tâche cron pour exécuter une fois par semaine une application PHP implémentée par l'utilisateur et qui collecte les données des ventes à partir du serveur et envoie un rapport. Pour plus d'exemples d'utilisation d'une ressource cron, consultez cron.

```
cron "job_name" do
  hour "1"
  minute "10"
  weekday "6"
  command "cd /srv/www/myapp/current && php .lib/mailing.php"
end
```

`cron` est une ressource Chef qui représente une tâche `cron`. Lorsque AWS OpsWorks Stacks exécute la recette sur une instance, le fournisseur associé gère les détails de configuration de la tâche.

- *`job_name`* est un nom défini par l'utilisateur pour la tâche `cron`, tel que `weekly report`.
- `hour`/`minute`/`weekday` spécifie à quel moment les commandes doivent être exécutées. Cet exemple exécute les commandes chaque samedi à 1 h 10.
- `command` spécifie les commandes à exécuter.

  Cet exemple exécute deux commandes. La première accède au répertoire `/srv/www/myapp/current`. La seconde exécute l'application `mailing.php` implémentée par l'utilisateur et qui collecte les données des ventes et envoie le rapport.

> **Note**
>
> La commande `bundle` ne fonctionne pas avec les tâches `cron` par défaut. La raison en est que AWS OpsWorks Stacks installe le bundler dans le répertoire. `/usr/local/bin` Pour utiliser `bundle` avec une tâche `cron`, vous devez ajouter explicitement le chemin d'accès `/usr/local/bin` à la tâche cron. En outre, comme la variable d'environnement $PATH peut ne pas se développer dans la tâche `cron`, une bonne pratique consiste à ajouter

explicitement à la tâche les informations de chemin d'accès nécessaires sans s'appuyer sur l'expansion de la variable $PATH. Les exemples suivants illustrent deux façons d'utiliser `bundle` dans une tâche `cron`.

```
cron "my first task" do
  path "/usr/local/bin"
  minute "*/10"
  command "cd /srv/www/myapp/current && bundle exec my_command"
end
```

```
cron_env = {"PATH" => "/usr/local/bin"}
cron "my second task" do
  environment cron_env
  minute "*/10"
  command "cd /srv/www/myapp/current && /usr/local/bin/bundle exec my_command"
end
```

Si votre stack comporte plusieurs serveurs d'applications, l'attribution des `cronjob.rb` événements du cycle de vie de la couche PHP App Server n'est peut-être pas une approche idéale. Par exemple, comme la recette s'exécute sur toutes les instances de la couche, vous recevrez plusieurs rapports. Une meilleure approche consiste à utiliser une couche personnalisée pour s'assurer qu'un seul serveur envoie un rapport.

Pour exécuter une recette sur une seule des instances d'une couche

1. Créez une couche personnalisée appelée, par exemple, PHPAdmin et attribuez-la `cronjob.rb` à ses événements de configuration et de déploiement. Les couches personnalisées n'ont pas nécessairement à en faire beaucoup. Dans ce cas, PHPAdmin il suffit d'exécuter une recette personnalisée sur ses instances.
2. Attribuez l'une des instances de PHP App Server à AdminLayer. Si une instance appartient à plusieurs couches, AWS OpsWorks Stacks exécute les recettes intégrées et personnalisées de chaque couche.

Comme une seule instance appartient au serveur d'applications PHP et aux PHPAdmin couches, elle ne `cronjob.rb` s'exécute que sur cette instance et vous ne recevez qu'un seul rapport.

Installation et configuration des packages sur les instances Linux

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les couches intégrées ne prennent en charge que certains packages. Pour de plus amples informations, veuillez consulter Couches. Vous pouvez installer d'autres packages, tels qu'un serveur Redis, en implémentant des recettes personnalisées pour gérer les tâches d'installation, de configuration de déploiement associées. Dans certains cas, la meilleure approche consiste à étendre une couche intégrée afin que le package soit installé sur ses instances en même temps que les packages standard de la couche. Par exemple, si vous avez une pile qui prend en charge une application PHP et que vous souhaitez inclure un serveur Redis, vous pouvez étendre la couche PHP App Server pour installer et configurer un serveur Redis sur les instances de la couche en plus d'un serveur d'applications PHP.

Une recette d'installation de package doit généralement exécuter des tâches telles que celles-ci :

- Créer un ou plusieurs répertoires et définir leurs modes.
- Créer un fichier de configuration à partir d'un modèle.
- Exécuter le programme d'installation pour installer le package sur l'instance.
- Démarrer un ou plusieurs services.

Pour obtenir un exemple d'installation du serveur Tomcat, consultez Création d'une couche serveur Tomcat personnalisée. La rubrique explique comment configurer une couche Redis personnalisée, mais vous pouvez utiliser pratiquement le même code pour installer et configurer Redis sur une couche intégrée. Pour des exemples d'installation d'autres packages, consultez les livres de recettes intégrés, sur https://github.com/aws/opsworks-cookbooks.

Création d'une couche serveur Tomcat personnalisée

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette rubrique explique comment implémenter une couche personnalisée pour une pile Linux. Cependant, les principes de base et une partie du code peuvent aussi être adaptés afin d'implémenter des couches personnalisées pour les piles Windows, notamment dans la section relative au déploiement d'applications.

Le moyen le plus simple d'utiliser des packages non standard sur des instances AWS OpsWorks Stacks consiste à étendre une couche existante. Cependant, cette approche installe et exécute aussi bien les packages standard que les packages non standard sur les instances de la couche, ce qui n'est pas toujours souhaitable. Une approche plus exigeante, mais plus efficace, consiste à implémenter une couche personnalisée, laquelle vous offre un contrôle presque complet sur les instances de la couche, éléments suivants inclus :

- Packages installés
- Configuration de chaque package
- Déploiement des applications d'un référentiel sur l'instance

Que ce soit à l'aide de la console ou de l'API, vous créez et gérez une couche personnalisée tout comme une autre couche, comme décrit dans Couches personnalisées. Cependant, les recettes intégrées d'une couche personnalisée n'effectuent que quelques tâches très élémentaires, telles que l'installation d'un client Ganglia pour rapporter les métriques à un serveur maître Ganglia. Afin que les instances d'une couche personnalisée soient plus que fonctionnelles a minima, vous devez implémenter un ou plusieurs livres de recettes personnalisés avec les recettes Chef et les fichiers associés pour pouvoir gérer, entre autres, les tâches d'installation et de configuration de packages,

ou de déploiement d'applications. Néanmoins, vous n'avez pas nécessairement à tout implémenter de A à Z. Par exemple, si vous stockez les applications dans l'un des référentiels standard, vous pouvez utiliser les recettes de déploiement intégrées pour gérer la plus grande partie du travail d'installation des applications sur les instances de la couche.

> **Note**
>
> Si vous utilisez Chef pour la première fois, lisez Les bases des livres de recettes ; ce didacticiel présente les notions de base qui expliquent comment implémenter les livres de recettes et exécuter une grande variété de tâches courantes.

La procédure suivante décrit comment mettre en œuvre une couche personnalisée qui prend en charge un serveur d'applications Tomcat. La couche s'appuie sur un livre personnalisé nommé Tomcat, qui contient les recettes permettant, entre autres, de gérer l'installation et le déploiement de packages. La procédure pas à pas inclut des extraits du livre de recettes Tomcat. Vous pouvez télécharger le livre de recettes complet depuis son GitHub référentiel. Si vous ne maîtrisez pas Opscode Chef, lisez d'abord Livres de recettes et recettes.

> **Note**
>
> AWS OpsWorks Stacks inclut une couche Java App Server complète pour une utilisation en production. Comme le livre de recettes Tomcat a pour objectif d'expliquer l'implémentation des couches personnalisées, il ne prend en charge qu'une version limitée de Tomcat, d'où sont absentes certaines fonctionnalités telles que SSL. Pour obtenir un exemple d'implémentation complète, consultez le livre de recettes opsworks_java.

Le livre de recettes Tomcat prend en charge une couche personnalisée dont les instances possèdent les caractéristiques suivantes :

- Elles prennent en charge un serveur d'applications Java Tomcat avec un serveur frontal Apache.
- Tomcat est configuré pour permettre aux applications d'utiliser un `DataSource` objet JDBC pour se connecter à une instance MySQL distincte, qui sert de magasin de données principal.

Le livre de recettes du projet comporte plusieurs composants clés :

- Le fichier d'attributs contient les paramètres de configuration utilisés par les différentes recettes.

- Les recettes Setup sont affectées à l'événement de cycle de vie Setup de la couche. Elles s'exécutent après le démarrage d'une instance et accomplissent différentes tâches telles que l'installation de packages et la création de fichiers de configuration.
- Les recettes Configure sont affectées à l'événement de cycle de vie Configure de la couche. Ils s'exécutent après les modifications de configuration de la pile, principalement lorsque les instances sont mises en ligne ou hors ligne, et gèrent les modifications de configuration requises.
- Les recettes Deploy sont attribuées à l'événement de cycle de vie Deploy de la couche. Elles s'exécutent après les recettes Setup et lorsque vous déployez manuellement une application pour installer le code et les fichiers associés sur les instances de la couche et gérer les tâches connexes, telles que le redémarrage des services.

La dernière section décrit comment créer une pile qui inclut une couche personnalisée basée sur le livre de recettes Tomcat et comment déployer et exécuter une application JSP simple qui affiche les données d'une base de données MySQL exécutée sur une instance appartenant à une couche MySQL distincte. Créer une pile et exécuter une application

> **Note**
>
> Les recettes du livre de recettes Tomcat dépendent de certaines recettes intégrées à AWS OpsWorks Stacks. Pour que l'origine de chaque recette apparaisse clairement, la présente rubrique identifie les recettes selon la convention Chef nom_du_livre_de_recettes::nom_de_recette.

Rubriques

- Fichiers d'attributs
- Recettes Setup
- Recettes Configure
- Recettes Deploy
- Créer une pile et exécuter une application

Fichiers d'attributs

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Avant de regarder les recettes, il est utile d'examiner le fichier d'attributs du livre de recettes Tomcat. Ce fichier contient différents paramètres de configuration utilisés par les recettes. Les attributs ne sont pas obligatoires ; vous pouvez simplement coder ces valeurs en dur dans vos recettes ou modèles. Toutefois, si vous définissez les paramètres de configuration à l'aide d'attributs, vous pouvez utiliser la console ou l'API AWS OpsWorks Stacks pour modifier les valeurs en définissant des attributs JSON personnalisés, ce qui est plus simple et plus flexible que de réécrire le code de la recette ou du modèle chaque fois que vous souhaitez modifier un paramètre. Cette approche permet, par exemple, d'utiliser le même livre de recettes pour plusieurs piles, mais de configurer le serveur Tomcat différemment selon chaque pile. Pour de plus amples informations sur les attributs et leur remplacement, consultez Remplacement des attributs.

L'exemple suivant montre le fichier d'attributs complet, `default.rb`, qui se trouve dans le répertoire `attributes` du livre de recettes Tomcat.

```
default['tomcat']['base_version'] = 6
default['tomcat']['port'] = 8080
default['tomcat']['secure_port'] = 8443
default['tomcat']['ajp_port'] = 8009
default['tomcat']['shutdown_port'] = 8005
default['tomcat']['uri_encoding'] = 'UTF-8'
default['tomcat']['unpack_wars'] = true
default['tomcat']['auto_deploy'] = true
case node[:platform]
when 'centos', 'redhat', 'fedora', 'amazon'
  default['tomcat']['java_opts'] = ''
when 'debian', 'ubuntu'
  default['tomcat']['java_opts'] = '-Djava.awt.headless=true -Xmx128m -XX:
+UseConcMarkSweepGC'
```

```
end
default['tomcat']['catalina_base_dir'] = "/etc/tomcat#{node['tomcat']['base_version']}"
default['tomcat']['webapps_base_dir'] = "/var/lib/tomcat#{node['tomcat']
['base_version']}/webapps"
default['tomcat']['lib_dir'] = "/usr/share/tomcat#{node['tomcat']['base_version']}/lib"
default['tomcat']['java_dir'] = '/usr/share/java'
default['tomcat']['mysql_connector_jar'] = 'mysql-connector-java.jar'
default['tomcat']['apache_tomcat_bind_mod'] = 'proxy_http' # or: 'proxy_ajp'
default['tomcat']['apache_tomcat_bind_config'] = 'tomcat_bind.conf'
default['tomcat']['apache_tomcat_bind_path'] = '/tc/'
default['tomcat']['webapps_dir_entries_to_delete'] = %w(config log public tmp)
case node[:platform]
when 'centos', 'redhat', 'fedora', 'amazon'
  default['tomcat']['user'] = 'tomcat'
  default['tomcat']['group'] = 'tomcat'
  default['tomcat']['system_env_dir'] = '/etc/sysconfig'
when 'debian', 'ubuntu'
  default['tomcat']['user'] = "tomcat#{node['tomcat']['base_version']}"
  default['tomcat']['group'] = "tomcat#{node['tomcat']['base_version']}"
  default['tomcat']['system_env_dir'] = '/etc/default'
end
```

Les paramètres eux-mêmes sont présentés ultérieurement dans la section associée. En règle générale, les remarques suivantes s'appliquent :

- Comme toutes les définitions de nœud sont de type `default`, vous pouvez les remplacer par des attributs JSON personnalisés.
- Le fichier utilise une instruction `case` pour définir de manière conditionnelle certaines valeurs d'attribut en fonction du système d'exploitation de l'instance.

  Le nœud `platform` est généré par Ohai, outil de Chef, et représente le système d'exploitation de l'instance.

Recettes Setup

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les recettes Setup sont affectées à l'événement de cycle de vie Setup de la couche et s'exécutent une fois qu'une instance a démarré. Elles exécutent des tâches telles que l'installation de packages, la création de fichiers de configuration et le démarrage de services. Une fois l'exécution des recettes d'installation terminée, AWS OpsWorks Stacks exécute les recettes de déploiement pour déployer toutes les applications sur la nouvelle instance.

Rubriques

- tomcat::setup
- tomcat::install
- tomcat::service
- tomcat::container_config
- tomcat::apache_tomcat_bind

tomcat::setup

La recette `tomcat::setup` est destinée à être affectée à un événement du cycle de vie Setup de la couche.

```
include_recipe 'tomcat::install'
include_recipe 'tomcat::service'

service 'tomcat' do
  action :enable
end

# for EBS-backed instances we rely on autofs
bash '(re-)start autofs earlier' do
  user 'root'
  code <<-EOC
    service autofs restart
  EOC
  notifies :restart, resources(:service => 'tomcat')
end
```

```
include_recipe 'tomcat::container_config'
include_recipe 'apache2'
include_recipe 'tomcat::apache_tomcat_bind'
```

La recette `tomcat::setup` est en grande partie une méta-recette. Elle contient un ensemble de recettes dépendantes qui gèrent la plupart des détails de l'installation et de la configuration de Tomcat et des packages associés. La première partie de `tomcat::setup` exécute les recettes suivantes, présentées en détail ultérieurement :

- La recette tomcat::install installe le package du serveur Tomcat.
- La recette tomcat::service configure le service Tomcat.

La partie centrale de `tomcat::setup` active et démarre le service Tomcat :

- La ressource service de Chef active le service Tomcat au démarrage.
- La ressource Chef bash exécute un script Bash pour démarrer le démon autofs, qui est nécessaire pour les instances soutenues par Amazon EBS. La ressource demande ensuite à la ressource `service` de redémarrer le service Tomcat.

  Pour plus d'informations, consultez autofs (pour Amazon Linux) ou Autofs (pour Ubuntu).

La partie finale de `tomcat::setup` crée les fichiers de configuration, puis installe et configure le serveur frontal Apache :

- La recette tomcat::container_config crée les fichiers de configuration.
- La `apache2` recette (qui est un raccourci pour`apache2::default`) est une recette intégrée à AWS OpsWorks Stacks qui installe et configure un serveur Apache.
- La recette tomcat::apache_tomcat_bind configure le serveur Apache pour qu'il fonctionne comme serveur frontal pour le serveur Tomcat.

> **Note**
>
> Vous pouvez souvent économiser du temps et des efforts en utilisant les recettes intégrées pour exécuter certaines tâches requises. Cette recette utilise la recette intégrée `apache2::default` pour installer Apache plutôt que de l'implémenter à partir de rien. Pour obtenir un autre exemple de l'utilisation des recettes intégrées, consultez Recettes Deploy.

Les sections suivantes décrivent plus en détail les recettes Setup du livre de recettes Tomcat. Pour plus d'informations sur les recettes `apache2`, consultez opsworks-cookbooks/apache2.

tomcat::install

La recette `tomcat::install` installe le serveur Tomcat, OpenJDK et une bibliothèque de connecteurs Java qui gère la connexion au serveur MySQL.

```
tomcat_pkgs = value_for_platform(
  ['debian', 'ubuntu'] => {
    'default' => ["tomcat#{node['tomcat']['base_version']}", 'libtcnative-1',
 'libmysql-java']
  },
  ['centos', 'redhat', 'fedora', 'amazon'] => {
    'default' => ["tomcat#{node['tomcat']['base_version']}", 'tomcat-native', 'mysql-
connector-java']
  },
  'default' => ["tomcat#{node['tomcat']['base_version']}"]
)

tomcat_pkgs.each do |pkg|
  package pkg do
    action :install
  end
end

link ::File.join(node['tomcat']['lib_dir'], node['tomcat']['mysql_connector_jar']) do
  to ::File.join(node['tomcat']['java_dir'], node['tomcat']['mysql_connector_jar'])
  action :create
end

# remove the ROOT webapp, if it got installed by default
include_recipe 'tomcat::remove_root_webapp'
```

La recette exécute les tâches suivantes :

1. Crée une liste de packages à installer, en fonction du système d'exploitation de l'instance.
2. Installe chaque package de la liste.

   La ressource du package Chef utilise le fournisseur approprié (`yum`pour Amazon Linux et `apt-get` pour Ubuntu) pour gérer l'installation. Les fournisseurs de package installent OpenJDK

comme dépendance Tomcat, mais la bibliothèque de connecteurs MySQL doit être installée explicitement.

3. Utilise une ressource link de Chef pour créer un lien symbolique entre le répertoire lib du serveur Tomcat et la bibliothèque de connecteurs MySQL du JDK.

   Selon les valeurs d'attribut par défaut, le répertoire lib Tomcat est `/usr/share/tomcat6/lib` et la bibliothèque de connecteurs MySQL (`mysql-connector-java.jar`) se trouve dans `/usr/share/java/`.

La recette `tomcat::remove_root_webapp` supprime l'application web racine (`/var/lib/tomcat6/webapps/ROOT` par défaut) pour éviter certains problèmes de sécurité.

```
ruby_block 'remove the ROOT webapp' do
  block do
    ::FileUtils.rm_rf(::File.join(node['tomcat']['webapps_base_dir'], 'ROOT'), :secure
 => true)
  end
  only_if { ::File.exists?(::File.join(node['tomcat']['webapps_base_dir'], 'ROOT'))
 && !::File.symlink?(::File.join(node['tomcat']['webapps_base_dir'], 'ROOT')) }
end
```

L'instruction `only_if` garantit que la recette ne supprime le fichier que s'il existe.

> **Note**
>
> La version Tomcat est spécifiée par l'attribut `['tomcat']['base_version']`, défini sur 6 dans le fichier d'attributs. Pour installer Tomcat 7, vous pouvez utiliser des attributs JSON personnalisés pour remplacer l'attribut. Modifiez simplement vos paramètres de pile et entrez le code JSON suivant dans la zone Custom Chef JSON (JSON Chef personnalisé), ou ajoutez-le à un JSON personnalisé existant :
>
> ```
> {
>   'tomcat' : {
>     'base_version' : 7
>   }
> }
> ```

L'attribut JSON personnalisé remplace l'attribut par défaut et définit la version Tomcat sur 7. Pour plus d'informations sur le remplacement des attributs, consultez Remplacement des attributs.

tomcat::service

La recette `tomcat::service` crée la définition du service Tomcat.

```
service 'tomcat' do
  service_name "tomcat#{node['tomcat']['base_version']}"

  case node[:platform]
  when 'centos', 'redhat', 'fedora', 'amazon'
    supports :restart => true, :reload => true, :status => true
  when 'debian', 'ubuntu'
    supports :restart => true, :reload => false, :status => true
  end

  action :nothing
end
```

La recette utilise la ressource service de Chef pour spécifier le nom de service Tomcat (tomcat6, par défaut) et définit l'attribut `supports` pour définir la façon dont Chef gère le redémarrage du service, le rechargement et les commandes d'état sur les différents systèmes d'exploitation.

- `true` indique que Chef peut utiliser le script init ou un autre fournisseur de services pour exécuter la commande.
- `false` indique que le Chef doit tenter d'exécuter la commande par d'autres moyens.

Notez que l'élément `action` est défini sur `:nothing`. Pour chaque événement du cycle de vie, AWS OpsWorks Stacks lance une exécution Chef pour exécuter l'ensemble de recettes approprié. Le livre de recettes Tomcat suit un modèle commun, selon lequel une recette crée la définition de service, mais ne redémarre pas le service. D'autres recettes de l'exécution Chef gèrent le redémarrage, généralement en incluant une commande `notifies` dans les ressources `template` utilisées pour créer les fichiers de configuration. Les notifications sont un moyen pratique de redémarrer un service, car elle ne le font que si la configuration a changé. En outre, si une exécution Chef comporte

plusieurs notifications de redémarrage pour un service, Chef redémarre le service au plus une fois. Cette pratique évite les problèmes susceptibles de se produire lorsque vous tentez de redémarrer un service qui n'est pas pleinement opérationnel, ce qui est une source commune d'erreurs Tomcat.

Le service Tomcat doit être défini pour toute exécution Chef qui utilise les notifications de redémarrage. `tomcat::service` est par conséquent inclus dans plusieurs recettes afin de garantir que le service soit défini pour chaque exécution Chef. Aucune pénalité ne s'applique si une exécution Chef inclut plusieurs instances de `tomcat::service`, car Chef garantit qu'une recette ne s'exécute qu'une seule fois par exécution Chef, quel que soit le nombre de fois où elle est incluse.

tomcat::container_config

La recette `tomcat::container_config` crée les fichiers de configuration à partir des fichiers modèles du livre de recettes.

```
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
```

La première recette appelle `tomcat::service`, qui définit le service si nécessaire. Le bloc principal de la recette se compose de deux ressources template, chacune d'elles créant un fichier de

configuration à partir de l'un des fichiers modèles du livre de recettes, définissant les propriétés du fichier et informant Chef de redémarrer le service.

Fichier de configuration de l'environnement Tomcat

La première ressource `template` utilise le fichier modèle `tomcat_env_config.erb` pour créer un fichier de configuration de l'environnement Tomcat, qui permet de définir les variables d'environnement telles que `JAVA_HOME`. Le nom de fichier par défaut est l'argument de la ressource `template`. `tomcat::container_config` utilise un attribut `path` pour remplacer la valeur par défaut et nommer le fichier de configuration `/etc/sysconfig/tomcat6` (Amazon Linux) ou `/etc/default/tomcat6` (Ubuntu). La ressource `template` spécifie également le propriétaire du fichier, le groupe et le mode, et demande à Chef de ne pas créer de fichiers de sauvegarde.

Si vous regardez le code source, il existe de fait trois versions de `tomcat_env_config.erb`, chacune dans un sous-répertoire différent du répertoire `templates`. Les répertoires `ubuntu` et `amazon` contiennent les modèles de leurs systèmes d'exploitation respectifs. Le dossier `default` contient un modèle vide avec une seule ligne de commentaire, qui n'est utilisée que si vous tentez d'exécuter cette recette sur une instance dont le système d'exploitation n'est pas pris en charge. La recette `tomcat::container_config` n'a pas besoin de spécifier le modèle `tomcat_env_config.erb` à utiliser. Chef sélectionne automatiquement le répertoire approprié pour le système d'exploitation de l'instance en fonction des règles décrites dans File Specificity.

Les fichiers `tomcat_env_config.erb` de cet exemple se composent essentiellement de commentaires. Pour définir des variables d'environnement supplémentaires, supprimez les marques de commentaire des lignes appropriées et fournissez vos valeurs préférées.

> **Note**
>
> Tout paramètre de configuration susceptible de changer doit être défini comme attribut plutôt que codé en dur dans le modèle. De cette façon, vous n'avez pas à réécrire le modèle pour modifier un paramètre et il vous suffit de remplacer l'attribut.

Le modèle Amazon Linux définit une seule variable d'environnement, comme illustré dans l'extrait suivant.

```
...
# 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 permet de spécifier des options Java telles que le chemin d'accès de la bibliothèque. A l'aide des valeurs d'attribut par défaut, le modèle ne définit aucune option Java pour Amazon Linux. Vous pouvez définir vos propres options Java en remplaçant l'attribut `['tomcat']['java_opts']`, à l'aide, par exemple, des attributs JSON personnalisés. Pour obtenir un exemple, consultez Créez une pile.

Le modèle Ubuntu définit plusieurs variables d'environnement, comme illustré dans l'extrait de modèle suivant.

```
# 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 -%>
```

A l'aide des valeurs d'attribut par défaut, le modèle définit les variables d'environnement Ubuntu comme suit :

- `TOMCAT6_USER` et `TOMCAT6_GROUP`, qui représentent le groupe et l'utilisateur Tomcat, sont tous deux définis sur `tomcat6`.

Si vous définissez ['tomcat']['base_version'] sur `tomcat7`, les noms des variables sont résolus en `TOMCAT7_USER` et `TOMCAT7_GROUP`, et les deux sont définis sur `tomcat7`.

- `JAVA_OPTS` a la valeur `-Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC` :
  - L'attribution à `-Djava.awt.headless` de la valeur `true` informe le moteur graphique que l'instance est sans périphérique de contrôle et sans console, ce qui prend en compte le comportement défaillant de certaines applications graphiques.
  - `-Xmx128m` garantit que la machine virtuelle Java dispose des ressources mémoire adéquates (128 Mo, dans cet exemple).
  - `-XX:+UseConcMarkSweepGC` spécifie le nettoyage de la mémoire « mark and sweep » simultané, ce qui aide à limiter les interruptions induites par le nettoyage de la mémoire.

    Pour plus d'informations, consultez : Concurrent Mark Sweep Collector Enhancements.
- Si la version Tomcat est inférieure à la version 7, le modèle annule la définition de `LC_ALL`, ce qui résout un bogue Ubuntu.

> **Note**
>
> Avec les attributs par défaut, certaines variables d'environnement sont simplement définies sur leurs valeurs par défaut. Cependant, la définition explicite des variables d'environnement avec les attributs signifie que vous pouvez définir des attributs JSON personnalisés pour remplacer les attributs par défaut et fournir des valeurs personnalisées. Pour plus d'informations sur le remplacement des attributs, consultez Remplacement des attributs.

Pour les fichiers modèles complets, consultez le code source.

Fichier de configuration Server.xml

La deuxième `template` ressource est utilisée `server.xml.erb` pour créer le fichier `system.xml` de configuration, qui configure le servlet/JSP conteneur. `server.xml.erb`ne contient aucun paramètre spécifique au système d'exploitation, il se trouve donc dans le `template` sous-répertoire du `default` répertoire.

Le modèle utilise les paramètres standard, mais peut créer un fichier `system.xml` pour Tomcat 6 ou Tomcat 7. Par exemple, le code suivant de la section serveur du modèle configure les écouteurs de façon appropriée pour la version spécifiée.

```
<% 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 -%>
```

Comme le modèle utilise des attributs à la place de paramètres codés en dur, vous pouvez facilement modifier les paramètres en définissant des attributs JSON personnalisés. Par exemple :

```
<Connector port="<%= node['tomcat']['port'] %>" protocol="HTTP/1.1"
           connectionTimeout="20000"
           URIEncoding="<%= node['tomcat']['uri_encoding'] %>"
           redirectPort="<%= node['tomcat']['secure_port'] %>" />
```

Pour plus d'informations, consultez le code source.

tomcat::apache_tomcat_bind

La recette `tomcat::apache_tomcat_bind` permet au serveur Apache de faire office de serveur frontal de Tomcat : il reçoit les demandes entrantes, les transmet à Tomcat et retourne les réponses au client. Cet exemple utilise mod_proxy comme proxy/passerelle Apache.

```
execute 'enable mod_proxy for apache-tomcat binding' do
  command '/usr/sbin/a2enmod proxy'
  not_if do
    ::File.symlink?(::File.join(node['apache']['dir'], 'mods-enabled', 'proxy.load'))
 || node['tomcat']['apache_tomcat_bind_mod'] !~ /\Aproxy/
  end
end

execute 'enable module for apache-tomcat binding' do
  command "/usr/sbin/a2enmod #{node['tomcat']['apache_tomcat_bind_mod']}"
  not_if {::File.symlink?(::File.join(node['apache']['dir'], 'mods-enabled',
 "#{node['tomcat']['apache_tomcat_bind_mod']}.load"))}
end

include_recipe 'apache2::service'

template 'tomcat thru apache binding' do
  path ::File.join(node['apache']['dir'], 'conf.d', node['tomcat']
['apache_tomcat_bind_config'])
  source 'apache_tomcat_bind.conf.erb'
  owner 'root'
  group 'root'
  mode 0644
  backup false
  notifies :restart, resources(:service => 'apache2')
end
```

Pour activer `mod_proxy`, vous devez activer le module `proxy` et un module basé sur le protocole. Vous avez deux options pour le module de protocole :

- HTTP: `proxy_http`
- JServ Protocole Apache (AJP) : `proxy_ajp`

  AJP est un protocole Tomcat interne.

Les deux ressources execute de la recette exécutent la commande `a2enmod`, qui active le module spécifié en créant les liens symboliques requis :

- La première ressource `execute` exécute le module `proxy`.
- La seconde ressource `execute` active le module de protocole, défini sur `proxy_http` par défaut.

Si vous préférez utiliser AJP, vous pouvez définir un JSON personnalisé pour remplacer l'attribut `apache_tomcat_bind_mod` et le définir sur `proxy_ajp`.

La `apache2::service` recette est une recette intégrée à AWS OpsWorks Stacks qui définit le service Apache. Pour plus d'informations, consultez la recette dans le GitHub référentiel AWS OpsWorks Stacks.

La ressource `template` utilise `apache_tomcat_bind.conf.erb` pour créer un fichier de configuration, nommé `tomcat_bind.conf` par défaut. Elle place le fichier dans le répertoire `['apache']['dir']/.conf.d`. L'attribut `['apache']['dir']` est défini dans le fichier d'attributs `apache2` intégré et est défini par défaut sur `/etc/httpd` (Amazon Linux) ou `/etc/apache2` (Ubuntu). Si la ressource `template` crée ou modifie le fichier de configuration, la commande `notifies` planifie un redémarrage du service Apache.

```
<% if node['tomcat']['apache_tomcat_bind_mod'] == 'proxy_ajp' -%>
ProxyPass <%= node['tomcat']['apache_tomcat_bind_path'] %> ajp://localhost:<%=
 node['tomcat']['ajp_port'] %>/
ProxyPassReverse <%= node['tomcat']['apache_tomcat_bind_path'] %> ajp://localhost:<%=
 node['tomcat']['ajp_port'] %>/
<% else %>
ProxyPass <%= node['tomcat']['apache_tomcat_bind_path'] %> http://localhost:<%=
 node['tomcat']['port'] %>/
ProxyPassReverse <%= node['tomcat']['apache_tomcat_bind_path'] %> http://localhost:<%=
 node['tomcat']['port'] %>/
<% end -%>
```

Le modèle utilise les ProxyPassReversedirectives ProxyPasset pour configurer le port utilisé pour transmettre le trafic entre Apache et Tomcat. Comme les deux serveurs sont sur la même instance, ils peuvent utiliser une URL localhost et sont tous deux définis par défaut sur `http://localhost:8080`.

Recettes Configure

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les recettes Configure sont affectées à l'événement de cycle de vie Configure de la recette, qui se produit sur toutes les instances de la pile chaque fois qu'une instance entre ou quitte l'état « en ligne » (online). Vous utilisez les recettes Configure pour ajuster la configuration d'une instance afin de répondre à la modification, le cas échéant. Lorsque vous implémentez une recette Configure, gardez à l'esprit qu'une modification de configuration de la pile peut impliquer des instances qui n'ont rien à voir avec la couche. La recette doit être capable de répondre de manière appropriée, ce qui peut signifier de ne rien faire dans certains cas.

tomcat::configure

La recette `tomcat::configure` est destinée à l'événement de cycle de vie Configure d'une couche.

```
include_recipe 'tomcat::context'
# Optional: Trigger a Tomcat restart in case of a configure event, if relevant
# settings in custom JSON have changed (e.g. java_opts/JAVA_OPTS):
#include_recipe 'tomcat::container_config'
```

La recette `tomcat::configure` est en gros une méta-recette qui exécute deux recettes dépendantes.

1. La recette `tomcat::context` crée un fichier de configuration du contexte de l'application web.

   Ce fichier configure les ressources JDBC que les applications utilisent pour communiquer avec l'instance MySQL, comme indiqué dans la section suivante. L'exécution de cette recette en réponse à un événement Configure permet à la couche de mettre à jour le fichier de configuration du contexte de l'application web si la couche base de données a été modifiée.
2. La recette Setup `tomcat::container_config` est exécutée à nouveau pour capturer les modifications de la configuration du conteneur.

Le code `include` pour `tomcat::container_config` est mis en commentaire dans cet exemple. Si vous souhaitez utiliser un JSON personnalisé pour modifier les paramètres Tomcat, vous pouvez supprimer le commentaire. Un événement du cycle de vie Configure exécute alors

`tomcat::container_config`, qui met à jour les fichiers de configuration Tomcat associés, comme décrit dans tomcat::container_config et redémarre le service Tomcat.

tomcat::context

Le livre de recettes Tomcat permet aux applications d'accéder à un serveur de base de données MySQL, qui peut être exécuté sur une instance distincte, à l'aide d'un objet DataSourceJ2EE. Avec Tomcat, vous pouvez activer la connexion en créant et en installant un fichier de configuration du contexte de l'application web pour chaque application. Ce fichier définit la relation entre l'application et la ressource JDBC que l'application utilise pour communiquer avec la base de données. Pour plus d'informations, consultez The Context Container.

La recette `tomcat::context` a pour objectif principal de créer ce fichier de configuration.

```
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
```

Outre les attributs du livre de recettes Tomcat, cette recette utilise les attributs de configuration et de déploiement de la pile installés par AWS OpsWorks Stacks avec l'événement Configure. Le service AWS OpsWorks Stacks ajoute des attributs à l'objet nœud de chaque instance qui contiennent les informations que les recettes obtiendraient généralement en utilisant des sacs de données ou en effectuant une recherche, et installe les attributs sur chaque instance. Les attributs contiennent des informations détaillées sur la configuration de la pile, les applications déployées

et les données personnalisées qu'un utilisateur souhaite inclure. Les recettes peuvent obtenir les données à partir des attributs de configuration et de déploiement de pile en utilisant la syntaxe de nœud Chef standard. Pour de plus amples informations, veuillez consulter Attributs de déploiement et de configuration de pile. Avec les piles Chef 11.10, vous pouvez également utiliser la recherche Chef pour obtenir les données de configuration et de déploiement de pile. Pour de plus amples informations, veuillez consulter Utilisation de Chef Search.

`deploy`attributes fait référence à l'`[:deploy]`espace de noms, qui contient les attributs liés au déploiement définis via la console ou l'API, ou générés par le AWS OpsWorks service Stacks. L'attribut `deploy` inclut un attribut pour chaque application déployée, nommé d'après le nom court de l'application. Chaque attribut d'application contient un ensemble d'attributs qui caractérisent l'application, tels que la racine du document (`[:deploy][:`*`appname`*`][:document_root]`).

La recette `context` s'assure d'abord que le service est défini pour l'exécution Chef en appelant tomcat::service. Elle définit ensuite une variable `context_name` qui représente le nom du fichier de la configuration, à l'exception de l'extension `.xml`. Si vous utilisez la racine du document par défaut, `context_name` est défini comme nom court de l'application. Sinon, il est défini sur la racine du document spécifié. Comme l'exemple présenté dans Créer une pile et exécuter une application définit la racine du document sur `"ROOT"`, le contexte est ROOT et le fichier de configuration est nommé `ROOT.xml`.

La majeure partie de la recette parcourt la liste des applications déployées et, pour chaque application, utilise le modèle `webapp_context.xml.erb` pour créer un fichier de configuration du contexte. L'exemple ne déploie qu'une seule application, mais la définition de l'attribut `deploy` requiert que vous la traitiez comme une liste d'applications quelles qu'elles soient.

Comme le modèle `webapp_context.xml.erb` n'est pas spécifique au système d'exploitation, il se trouve dans le sous-répertoire `templates` du répertoire `default`.

La recette crée le fichier de configuration comme suit :

- A l'aide des valeurs d'attribut par défaut, le fichier de configuration est défini sur *`context_name`*`.xml` et installé dans le répertoire `/etc/tomcat6/Catalina/localhost/`.

  Le nœud `['datasources']` des attributs de configuration de la pile contient un ou plusieurs attributs, chacun d'eux mappant un nom de contexte et la ressource de données JDBC que l'application associée utilise pour communiquer avec la base de données. Le nœud et son contenu sont définis avec un JSON personnalisé lorsque vous créez la pile, comme décrit ultérieurement

dans Créer une pile et exécuter une application. L'exemple possède un attribut unique qui associe le nom de contexte ROOT à une ressource JDBC nommée jdbc/mydb.

- A l'aide des valeurs d'attribut par défaut, l'utilisateur du fichier et le groupe sont tous deux définis avec les valeurs définies par le package Tomcat : `tomcat` (Amazon Linux) ou `tomcat6` (Ubuntu).
- La ressource `template` ne crée le fichier de configuration que si le nœud `['datasources']`existe et inclut un attribut `context_name`.
- La ressource définit `template` deux variables, `resource_name` et `webapp_name`.

  `resource_name`est défini avec le nom de ressource associé à `context_name` et `webapp_name` est défini avec le nom court de l'application.
- La ressource modèle redémarre le service Tomcat pour charger et activer les modifications.

Le modèle `webapp_context.xml.erb` se compose d'un élément `Context` qui contient un élément `Resource` avec son propre ensemble d'attributs.

Les `Resource` attributs caractérisent la configuration du contexte :

- name — Le nom de la ressource JDBC, défini sur la `resource_name` valeur définie dans. `tomcat::context`

  Pour l'exemple, le nom de ressource est jdbc/mydb.
- auth et type : il s'agit des paramètres standard pour les connexions `DataSource` JDBC.
- MaxActive, MaxIdle et MaxWait : nombre maximal de connexions actives et inactives, et temps d'attente maximal pour qu'une connexion soit renvoyée.
- nom d'utilisateur et mot de passe : nom d'utilisateur et mot de passe root de la base de données, obtenus à partir des `deploy` attributs.
- driverClassName—Le nom de classe du pilote JDBC, défini sur le pilote MySQL.
- url —URL de connexion.

  Le préfixe dépend de la base de données. Il doit être défini sur `jdbc:mysql` pour MySQL, `jdbc:postgresql` pour Postgres et `jdbc:sqlserver` pour SQL Server. L'exemple définit l'URL sur`jdbc:mysql://`*`host_IP_Address`*`:3306:simplejsp`, où *`simplejsp`* est le nom abrégé de l'application.
- **`DataSource`**factory : la fabrique requise pour les bases de données MySQL.

Pour plus d'informations sur ce fichier de configuration, consultez la DataSources rubrique Utilisation du wiki Tomcat.

Recettes Deploy

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les recettes Deploy sont attribuées à l'événement de cycle de vie Deploy de la couche. Cela se produit généralement sur toutes les instances de la pile chaque fois que vous déployez une application, bien que vous puissiez éventuellement limiter l'événement aux instances spécifiées uniquement. AWS OpsWorks Stacks exécute également les recettes de déploiement sur les nouvelles instances, une fois les recettes de configuration terminées. Les recettes Deploy ont pour objectif principal de déployer le code et les fichiers associés d'un référentiel sur les instances de la couche serveur de l'application. Cependant, vous exécutez souvent les recettes sur d'autres couches également. Cela permet aux instances des couches de mettre à jour leur configuration, par exemple, pour accueillir l'application nouvellement déployée. Lorsque vous implémentez une recette Deploy, gardez à l'esprit qu'un événement Deploy ne signifie pas nécessairement que les applications sont déployées sur l'instance. Il peut s'agir d'une simple notification que les applications sont déployées sur d'autres instances de la pile pour permettre à l'instance effectuer les mises à jour nécessaires. La recette doit être capable de répondre de manière appropriée, ce qui implique de ne rien faire.

AWS OpsWorks Stacks déploie automatiquement les applications des types d'applications standard sur les couches de serveur d'applications intégrées correspondantes. Pour déployer les applications sur une couche personnalisée, vous devez implémenter les recettes Deploy personnalisées qui téléchargent les fichiers de l'application depuis un référentiel sur l'emplacement approprié de l'instance. Cependant, vous pouvez souvent limiter la quantité de code que vous devez écrire à l'aide du livre de recettes Deploy intégré pour gérer certains aspects de déploiement. Par exemple, si vous stockez vos fichiers dans l'un des référentiels pris en charge, le livre de recettes intégré peut gérer les détails du téléchargement des fichiers depuis le référentiel sur les instances de la couche.

La recette `tomcat::deploy` est destinée à être assignée à l'événement de cycle de vie Deploy.

```
include_recipe 'deploy'

node[:deploy].each do |application, deploy|
  opsworks_deploy_dir do
    user deploy[:user]
    group deploy[:group]
    path deploy[:deploy_to]
  end

  opsworks_deploy do
    deploy_data deploy
    app application
  end
...
```

La recette `tomcat::deploy` utilise le livre de recettes Deploy intégré pour les aspects du déploiement qui ne sont pas spécifiques à l'application. La recette `deploy` (qui est un raccourci de la recette `deploy::default` intégrée) est une recette intégrée qui gère les détails de la configuration des utilisateurs, groupes, etc., selon les données des attributs `deploy`.

La recette utilise deux définitions Chef intégrées, `opsworks_deploy_dir` et `opworks_deploy` pour installer l'application.

La définition `opsworks_deploy_dir` configure la structure du répertoire, à partir des données du déploiement JSON de l'application. Les définitions sont essentiellement un moyen pratique d'empaqueter les définitions des ressources et se trouvent dans le répertoire `definitions` d'un livre de recettes. Les recettes peuvent utiliser les définitions tout comme les ressources, mais la définition elle-même n'a pas de fournisseur associé, seules les ressources qui sont incluses dans la définition. Vous pouvez définir les variables dans la recette, qui sont transmises aux définitions de ressources sous-jacentes. La recette `tomcat::deploy` définit les variables `user`, `group` et `path` en fonction des données du déploiement JSON. Elles sont transmises à la ressource directory de la définition, qui gère les répertoires.

> **Note**
>
> Le groupe et l'utilisateur de votre application déployée sont déterminés par les attributs `[:opsworks][:deploy_user][:user]` et `[:opsworks][:deploy_user][:group]`, qui sont définis dans le fichier d'attributs `deploy.rb` du livre de recettes Deploy intégré. La valeur par défaut de `[:opsworks][:deploy_user][:user]` est `deploy`. La valeur par

défaut de `[:opsworks][:deploy_user][:group]` dépend du système d'exploitation de l'instance :

- Pour les instances Ubuntu, le groupe par défaut est `www-data`.
- Pour les instances Amazon Linux membres d'une couche Rails App Server utilisant Nginx et Unicorn, le groupe par défaut est. `nginx`
- Pour toutes les autres instances Amazon Linux, le groupe par défaut est `apache`.

Vous pouvez changer les deux paramètres à l'aide d'un JSON personnalisé ou d'un fichier d'attributs personnalisés pour remplacer l'attribut approprié. Pour de plus amples informations, veuillez consulter Remplacement des attributs.

L'autre définition, `opsworks_deploy`, gère les détails de la vérification du code de l'application et des fichiers associés du référentiel et de leur déploiement sur l'instance, selon les données des attributs `deploy`. Vous pouvez utiliser cette définition pour tout type d'application ; les détails du déploiement tels que les noms de répertoire sont spécifiés dans la console ou via l'API, et placés dans les attributs `deploy`. Toutefois, `opsworks_deploy` fonctionne uniquement pour les quatre types de référentiel pris en charge : Git, Subversion, S3 et HTTP. Vous devez implémenter ce code vous-même si vous souhaitez utiliser un autre type de référentiel.

Vous installez les fichiers d'une application dans le répertoire Tomcat `webapps`. Une pratique classique consiste à copier les fichiers directement sur `webapps`. Cependant, le déploiement de AWS OpsWorks Stacks est conçu pour conserver jusqu'à cinq versions d'une application sur une instance. Vous pouvez donc revenir à une version antérieure si nécessaire. AWS OpsWorks Stacks effectue donc ce qui suit :

1. Déploie les applications sur un répertoire distinct dont le nom contient un horodatage, comme `/srv/www/my_1st_jsp/releases/20130731141527`.
2. Crée un lien symbolique nommé `current`, tel que `/srv/www/my_1st_jsp/current`, avec ce répertoire unique.
3. S'il n'existe pas déjà, crée un lien symbolique depuis le répertoire `webapps` vers le lien symbolique `current` créé à l'étape 2.

Si vous devez revenir à une version antérieure, modifiez le lien symbolique `current` pour pointer sur un répertoire distinct contenant l'horodatage approprié en changeant, par exemple, la cible du lien de `/srv/www/my_1st_jsp/current`.

La section centrale de `tomcat::deploy` définit le lien symbolique.

```
  ...
  current_dir = ::File.join(deploy[:deploy_to], 'current')
  webapp_dir = ::File.join(node['tomcat']['webapps_base_dir'],
deploy[:document_root].blank? ? application : deploy[:document_root])

  # opsworks_deploy creates some stub dirs, which are not needed for typical webapps
  ruby_block "remove unnecessary directory entries in #{current_dir}" do
    block do
      node['tomcat']['webapps_dir_entries_to_delete'].each do |dir_entry|
        ::FileUtils.rm_rf(::File.join(current_dir, dir_entry), :secure => true)
      end
    end
  end

  link webapp_dir do
    to current_dir
    action :create
  end
  ...
```

La recette crée d'abord deux variables, `current_dir` et `webapp_dir`, pour représenter les répertoires `current` et `webapp`, respectivement. Elle utilise ensuite une ressource `link` pour lier `webapp_dir` à `current_dir`. La `deploy::default` recette AWS OpsWorks Stacks crée des répertoires stub qui ne sont pas obligatoires dans cet exemple, de sorte que la partie centrale de l'extrait les supprime.

La partie finale de `tomcat::deploy` redémarre le service Tomcat, si nécessaire.

```
  ...
  include_recipe 'tomcat::service'

  execute 'trigger tomcat service restart' do
    command '/bin/true'
    not_if { node['tomcat']['auto_deploy'].to_s == 'true' }
```

```
    notifies :restart, resources(:service => 'tomcat')
  end
end

include_recipe 'tomcat::context'
```

La recette exécute d'abord `tomcat::service`, afin de s'assurer que le service est défini pour cette exécution Chef. Elle utilise ensuite une ressource execute pour demander au service de redémarrer, mais seulement si `['tomcat']['auto_deploy']` est défini sur `'true'`. Sinon, Tomcat écoute les modifications dans son répertoire `webapps`, ce qui rend superflu le redémarrage explicite du service Tomcat.

> **Note**
>
> La ressource `execute` n'exécute pas réellement quelque chose de substantiel ; `/bin/true` est un script shell factice qui retourne simplement un code de réussite. Il est utilisé ici simplement comme moyen pratique de générer une notification de redémarrage. Comme mentionné précédemment, l'utilisation des notifications garantit que les services ne soient pas trop souvent redémarrés.

Enfin, `tomcat::deploy` exécute `tomcat::context`, qui met à jour le fichier de configuration du contexte de l'application web si vous avez modifié la base de données principale.

Créer une pile et exécuter une application

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section montre comment utiliser le livre de recettes Tomcat pour mettre en place une configuration de pile élémentaire qui exécute une simple application de pages serveur Java (JSP), nommée SimpleJSP. La pile se compose d'une couche personnalisée basée sur Tomcat nommée TomCustom et d'une couche MySQL. SimpleJSP est déployé sur la base de données MySQL

TomCustom et affiche certaines informations à partir de celle-ci. Si vous n'êtes pas déjà familiarisé avec les bases de l'utilisation de AWS OpsWorks Stacks, vous devez d'abord lireMise en route des piles Linux Chef 11.

Application SimpleJSP

L'application SimpleJSP illustre les bases de la configuration d'une connexion de base de données et de l'extraction des données à partir de la base de données MySQL de la pile.

```
<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 utilise un objet `DataSource` pour communiquer avec la base de données MySQL. Tomcat utilise les données du fichier de configuration du contexte de l'application web pour créer et initialiser un objet `DataSource` et le lier à un nom logique. Elle enregistre ensuite le nom logique avec un service d'attribution de noms JNDI (Java Naming et Directory Interface). Pour obtenir une instance de l'objet `DataSource` approprié, vous créez un objet `InitialContext` et passez le nom logique de la ressource à la méthode `lookup` de l'objet, qui récupère l'objet approprié. Le nom logique de l'exemple SimpleJSP, `java:comp/env/jdbc/mydb`, possède les éléments suivants :

- L'espace de noms racine, `java`, qui est séparé du reste du nom par le signe deux points (:).
- Les espaces de noms supplémentaires, séparés par des barres obliques (/).

  Tomcat ajoute automatiquement les ressources à l'espace de noms `comp/env`.
- Le nom de ressource, défini dans le fichier de configuration du contexte de l'application web et séparé des espaces de noms par une barre oblique.

  Le nom de la ressource pour cet exemple est `jdbc/mydb`.

Pour établir une connexion à la base de données, SimpleJSP effectue les opérations suivantes :

1. Appelle la méthode `DataSource` de l'objet `getConnection`, qui retourne un objet `Connection`.

2. Appelle la méthode `Connection` de l'objet `createStatement` pour créer un objet `Statement`, qui vous permet de communiquer avec la base de données.
3. Communique avec la base de données en appelant la méthode `Statement` appropriée.

   SimpleJSP appelle `executeQuery` pour exécuter une requête SHOW DATABASES, qui répertorie les bases de données du serveur.

La méthode `executeQuery` retourne un objet `ResultSet`, qui contient les résultats de la requête. SimpleJSP obtient les noms de base de données à partir de l'objet `ResultSet` retourné et les concatène pour créer une chaîne de sortie. Enfin, l'exemple ferme les objets `ResultSet`, `Statement` et `Connection`. Pour plus d'informations sur JSP et JDBC, consultez JavaServer Pages Technology et JDBC Basics, respectivement.

Pour utiliser SimpleJSP avec une pile, vous devez la placer dans un référentiel. Vous pouvez utiliser l'un des référentiels pris en charge, mais pour utiliser SimpleJSP avec l'exemple de pile présenté dans la section suivante, vous devez la placer dans une archive S3 publique. Pour plus d'informations sur l'utilisation des autres référentiels standard, consultez Référentiels de livres de recettes.

Pour placer SimpleJSP dans un référentiel d'archivage S3

1. Copiez l'exemple de code dans un fichier nommé `simplejsp.jsp` et placez le fichier dans un répertoire nommé `simplejsp`.
2. Créez une archive `.zip` du répertoire `simplejsp`.
3. Créez un compartiment Amazon S3 public, `simplejsp.zip` chargez-le dans le compartiment et rendez le fichier public.

   Pour obtenir une description de l'exécution de cette tâche, consultez Mise en route avec Amazon Simple Storage Service.

Créez une pile

Pour exécuter SimpleJSP, vous avez besoin d'une pile avec les couches suivantes.

- Une couche MySQL, qui prend en charge le serveur principal MySQL.
- Une couche personnalisée qui utilise le livre de recettes Tomcat pour prendre en charge les instances serveur Tomcat.

Pour créer la pile

1. Sur le tableau de bord AWS OpsWorks Stacks, cliquez sur Ajouter une pile pour créer une nouvelle pile, puis sur Avancé **>>** pour afficher toutes les options. Configurez la pile comme suit.

   - Nom : nom de pile défini par l'utilisateur ; cet exemple utilise TomStack.
   - Utiliser des livres de recettes Chef personnalisés : réglez le bouton sur Oui pour afficher des options supplémentaires.
   - Type de dépôt —Git.
   - URL du référentiel —`git://github.com/amazonwebservices/opsworks-example-cookbooks.git`.
   - Custom Chef JSON —Ajoutez le JSON suivant :

     ```
     {
       "tomcat": {
         "base_version": 7,
         "java_opts": "-Djava.awt.headless=true -Xmx256m"
       },
       "datasources": {
         "ROOT": "jdbc/mydb"
       }
     }
     ```

   Pour le reste des options, vous pouvez accepter les valeurs par défaut.

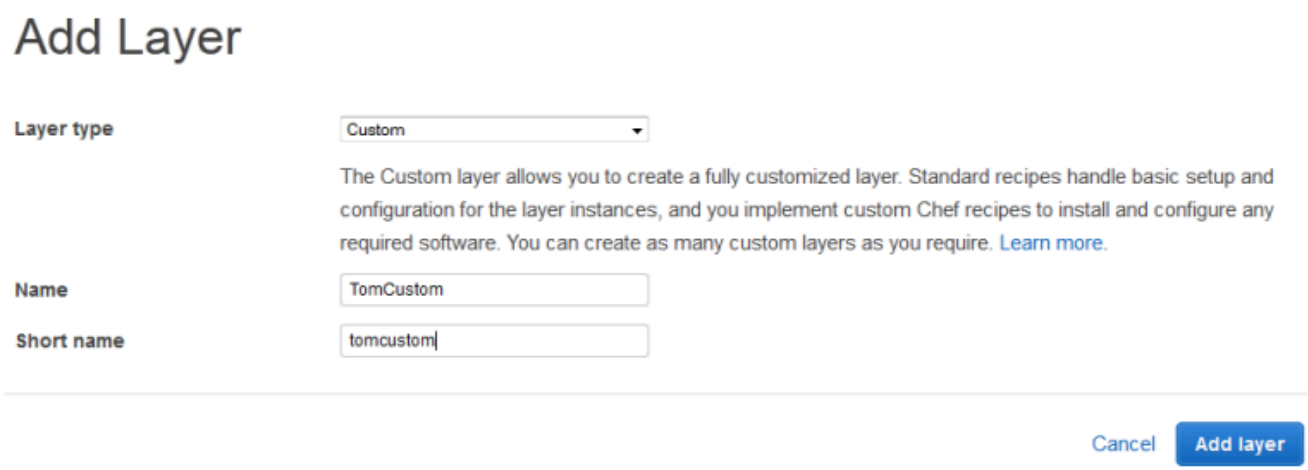
   Le JSON personnalisé exécute les tâches suivantes :

   - Remplace l'attribut `['base_version']` du livre de recettes Tomcat pour définir la version Tomcat sur 7 ; la valeur par défaut est 6.
   - Remplace l'attribut `['java_opts']` du livre de recettes Tomcat pour spécifier que l'instance est sans périphérique de contrôle et définir la taille maximale de segment de machine virtuelle Java sur 256 Mo ; la valeur par défaut ne définit aucune option pour les instances exécutant Amazon Linux.
   - Spécifie la valeur d'attribut `['datasources]`, qui attribue un nom de ressource JDBC (jdbc/mydb) au nom du contexte de l'application web (ROOT), comme indiqué dans tomcat::context.

Ce dernier attribut n'a aucune valeur par défaut ; vous devez la définir avec le JSON personnalisé.



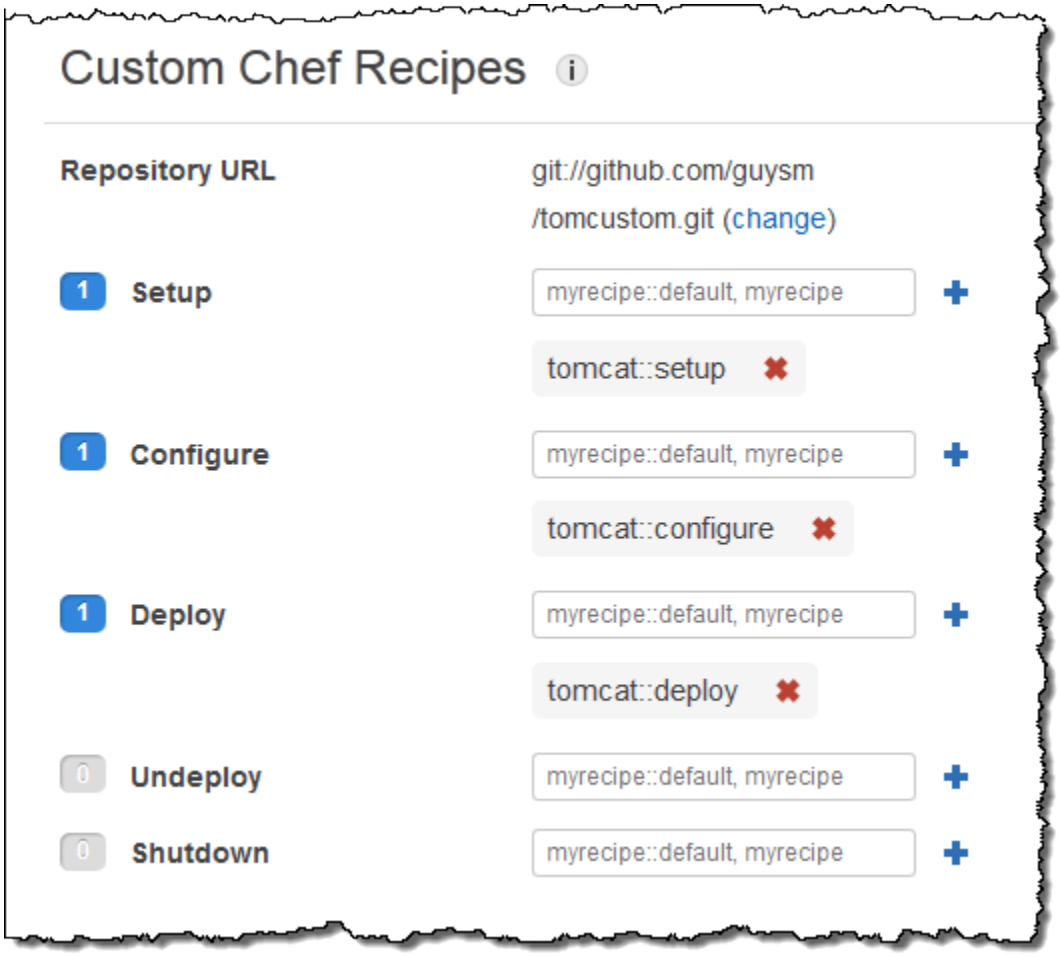
2. Cliquez sur Add a layer (Ajouter une couche). Pour Layer type (Type de couche), sélectionnez MySQL. Puis, cliquez sur Add Layer (Ajouter une couche).
3. Dans le panneau de navigation, cliquez sur Instances, puis cliquez sur Add an instance (Ajouter une instance). Cliquez sur Add Instance (Ajouter une instance) pour accepter les valeurs par défaut. Sur la ligne de l'instance, cliquez sur start (démarrer).
4. Revenez à la page Layers (Couches) et cliquez sur + Layer (+ Couche) pour ajouter une couche. Pour Layer type (Type de couche), cliquez sur Custom (Personnalisé). L'exemple utilise **`TomCustom`** et **`tomcustom`** comme nom de couche et nom court, respectivement.



5. Sur la page Layers (Couches), pour la couche personnalisée, cliquez sur Recipes (Recettes), puis cliquez sur Edit (Modifier). Sous Custom Chef Recipes (Recettes Chef personnalisées), affectez les recettes du livre de recettes Tomcat aux événements de cycle de vie de la couche, comme suit :

   - Pour Setup, tapez **`tomcat::setup`** et cliquez sur +.
   - Pour Configure, tapez **`tomcat::configure`** et cliquez sur +.
   - Pour Deploy, tapez **`tomcat::deploy`** et cliquez sur +. Puis, cliquez sur Save (Enregistrer).

   .



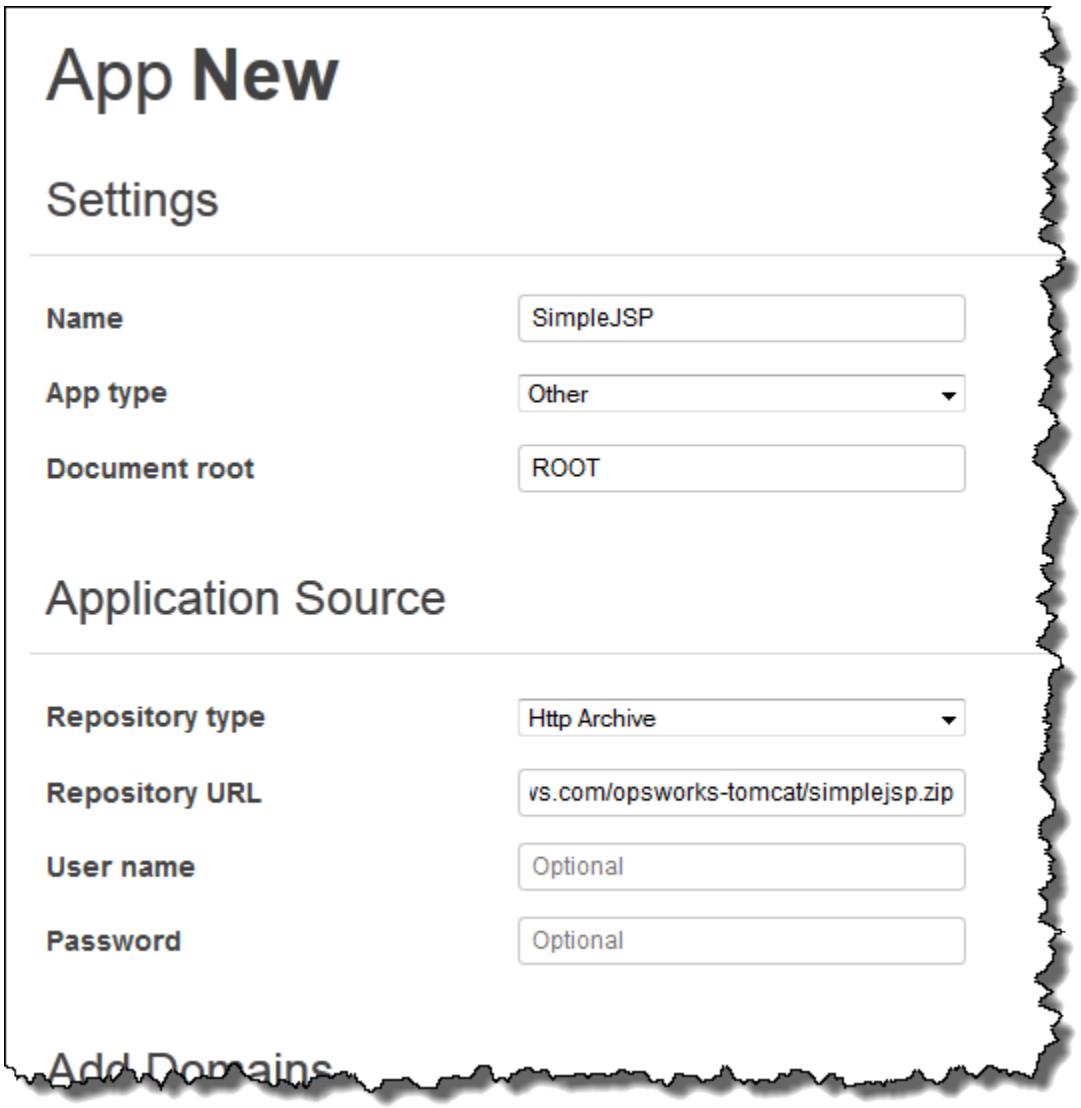
6. Dans le panneau de navigation, cliquez sur Apps (Applications), puis cliquez sur Add an app (Ajouter une application). Spécifiez les options suivantes, puis cliquez sur Add App (Ajouter une application) :

- Nom : nom de l'application ; l'exemple utilise SimpleJSP et le nom abrégé généré par AWS OpsWorks Stacks sera simplejsp.
- Type d'application : définissez cette option sur Autre.

  AWS OpsWorks Stacks déploie automatiquement les types d'applications standard sur les instances de serveur associées. Si vous définissez App type (Type d'application) avec la valeur Other, AWS OpsWorks Stacks exécute simplement les recettes Deploy et leur laisse gérer le déploiement.
- Racine du document —Définissez cette option sur. **`ROOT`**

  La valeur Document root (Racine du document) spécifie le nom du contexte.
- Type de référentiel : définissez cette option sur S3 Archive.
- URL du référentiel : définissez cette URL sur l'URL Amazon S3 de l'application que vous avez créée précédemment.

Utilisez les paramètres par défaut pour les autres options.



7. Utilisez la page Instances pour ajouter une instance à la TomCustom couche et la démarrer. AWS OpsWorks Stacks exécute automatiquement les recettes de déploiement sur une nouvelle instance une fois les recettes d'installation terminées. Le démarrage de l'instance déploie donc également SimpleJSP.
8. Lorsque l' TomCustom instance est en ligne, cliquez sur le nom de l'instance sur la page Instances pour voir ses détails. Copiez l'adresse IP publique. Créez ensuite une URL comme suit : http ://*publicIPappname.jsp*/tc/. Pour l'exemple, cette URL doit se présenter ainsi : **`http://50.218.191.172/tc/simplejsp.jsp`**.

> **Note**
>
> L'URL Apache qui achemine les requêtes vers Tomcat est définie avec l'attribut `['tomcat']['apache_tomcat_bind_path']` par défaut, `/tc/`. La racine du

document SimpleJSP est définie sur R00T, une valeur spéciale qui se résout en /. Par conséquent, l'URL est « .../tc/simplejsp.jsp ».

9. Collez l'URL de l'étape précédente dans votre navigateur. Vous devez voir ce qui suit :

```
Databases found:
information_schema
simplejsp
test
```

> **Note**
>
> Si votre stack possède une instance MySQL, AWS OpsWorks Stacks crée automatiquement une base de données pour chaque application, nommée avec le nom abrégé de l'application.

Attributs de déploiement et de configuration de pile

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Lorsque AWS OpsWorks Stacks exécute une commande sur une instance, par exemple une commande de déploiement en réponse à un événement du cycle de vie de déploiement, il ajoute un ensemble d'attributs à l'objet nœud de l'instance qui décrit la configuration actuelle de la pile. Pour les commandes Deploy events et Execute Recipes stack, AWS OpsWorks Stacks installe les attributs de déploiement, qui fournissent des informations de déploiement supplémentaires. Pour plus d'informations sur l'objet nœud, consultez Remplacement des attributs. Pour obtenir la liste des attributs de déploiement et de configuration de pile couramment utilisés, y compris les noms de nœud complets, consultez Attributs de déploiement et de configuration de pile : Linux et Attributs des livres de recettes intégrés.

> **Note**
>
> Sur les piles Linux, vous pouvez obtenir une liste complète de ces attributs, mis en forme comme un objet JSON, à l'aide de la commande get_json de l'interface de ligne de commande de l'agent.

Les sections suivantes décrivent les attributs associés à un événement Configure et à un événement Deploy d'une pile simple, qui comprend les éléments suivants :

- Une couche PHP App Server avec deux instances
- Une HAProxy couche avec une instance

Les exemples proviennent de l'une des instances de PHP App Server, php-app1. Pour plus de commodité, les attributs sont mis en forme comme un objet JSON. La structure de l'objet est mappé avec les noms complets des attributs. Par exemple, l'attribut `node[:opsworks][:ruby_version]` apparaît comme suit dans une représentation JSON.

```
{
  "opsworks": {
    ...
    "ruby_version": "1.8.7",
    ...
  }
}
```

Rubriques

- Attributs Configure
- Attributs de déploiement

Attributs Configure

L'objet JSON suivant illustre les attributs d'un événement Configure, qui se produit sur chaque instance de la pile lorsqu'une instance est mise en ligne ou hors connexion. Les attributs incluent les attributs intégrés de configuration de pile intégrée et les attributs JSON personnalisés qui ont été définis pour la pile avant l'événement (aucun dans cet exemple). Il a été modifié quant à la longueur.

Pour une description détaillée des différents attributs, consultez Attributs de déploiement et de configuration de pile : Linux et Attributs des livres de recettes intégrés.

```
{
  "opsworks": {
    "layers": {
      "php-app": {
        "id": "4a2a56c8-f909-4b39-81f8-556536d20648",
        "instances": {
          "php-app2": {
            "elastic_ip": null,
            "region": "us-west-2",
            "booted_at": "2013-02-26T20:41:10+00:00",
            "ip": "192.0.2.0",
            "aws_instance_id": "i-34037f06",
            "availability_zone": "us-west-2a",
            "instance_type": "c1.medium",
            "private_dns_name": "ip-10-252-0-203.us-west-2.compute.internal",
            "private_ip": "10.252.0.203",
            "created_at": "2013-02-26T20:39:39+00:00",
            "status": "online",
            "backends": 8,
            "public_dns_name": "ec2-192-0-2-0.us-west-2.compute.amazonaws.com"
          },
          "php-app1": {
            ...
          }
        },
        "name": "PHP Application Server"
      },
      "lb": {
        "id": "15c86142-d836-4191-860f-f4d310440f14",
        "instances": {
          "lb1": {
           ...
          }
        },
        "name": "Load Balancer"
      }
    },
    "agent_version": "104",
    "applications": [
```

```
    ],
    "stack": {
      "name": "MyStack"
    },
    "ruby_version": "1.8.7",
    "sent_at": 1361911623,
    "ruby_stack": "ruby_enterprise",
    "instance": {
      "layers": [
        "php-app"
      ],
      "region": "us-west-2",
      "ip": "192.0.2.0",
      "id": "45ef378d-b87c-42be-a1b9-b67c48edafd4",
      "aws_instance_id": "i-32037f00",
      "availability_zone": "us-west-2a",
      "private_dns_name": "ip-10-252-84-253.us-west-2.compute.internal",
      "instance_type": "c1.medium",
      "hostname": "php-app1",
      "private_ip": "10.252.84.253",
      "backends": 8,
      "architecture": "i386",
      "public_dns_name": "ec2-192-0-2-0.us-west-2.compute.amazonaws.com"
    },
    "activity": "configure",
    "rails_stack": {
      "name": null
    },
    "deployment": null,
    "valid_client_activities": [
      "reboot",
      "stop",
      "setup",
      "configure",
      "update_dependencies",
      "install_dependencies",
      "update_custom_cookbooks",
      "execute_recipes"
    ]
  },
  "opsworks_custom_cookbooks": {
    "recipes": [
```

```
    ],
    "enabled": false
  },
  "recipes": [
    "opsworks_custom_cookbooks::load",
    "opsworks_ganglia::configure-client",
    "ssh_users",
    "agent_version",
    "mod_php5_apache2::php",
    "php::configure",
    "opsworks_stack_state_sync",
    "opsworks_custom_cookbooks::execute",
    "test_suite",
    "opsworks_cleanup"
  ],
  "opsworks_rubygems": {
    "version": "1.8.24"
  },
  "ssh_users": {
  },
  "opsworks_bundler": {
    "manage_package": null,
    "version": "1.0.10"
  },
  "deploy": {
  }
}
```

La plupart des informations se trouvent sous l'attribut `opsworks`, qui est souvent désigné par l'appellation espace de noms. La liste suivante décrit les attributs clés :

- `layers`attributs : ensemble d'attributs, dont chacun décrit la configuration de l'une des couches de la pile.

  Les couches sont identifiées par leurs noms courts, `php-app` et `lb` dans cet exemple. Pour plus d'informations sur les noms courts des autres couches, consultez AWS OpsWorks Référence de la couche Stacks.
- `instances`attributs — Chaque couche possède un `instances` élément, qui inclut un attribut pour chacune des instances en ligne des couches, nommé avec le nom abrégé de l'instance.

  La couche PHP App Server possède deux instances, `php-app1` et`php-app2`. La HAProxy couche possède une instance,`lb1`.

> **Note**
>
> L'élément `instances` contient uniquement les instances qui sont dans l'état en ligne lorsque les attributs de configuration et de déploiement de pile sont créés.

- Attributs d'instance : chaque attribut d'instance contient un ensemble d'attributs qui caractérisent l'instance, tels que l'adresse IP privée et le nom DNS privé de l'instance. Pour des raisons de brièveté, l'exemple ne montre que l'attribut `php-app2` en détail ; les autres attributs contiennent des informations similaires.
- `applications`— Liste des applications déployées, non utilisées dans cet exemple.
- `stack`— Le nom de la pile ; `MyStack` dans cet exemple.
- `instance`— L'instance sur laquelle ces attributs sont installés ; `php-app1` dans cet exemple. Les recettes peuvent utiliser cet attribut pour obtenir des informations sur l'instance sur laquelle elles s'exécutent, telles que l'adresse IP publique de l'instance.
- `activity`— L'activité qui a produit les attributs ; un événement de configuration dans cet exemple.
- `rails_stack`— La pile Rails pour les piles qui incluent une couche Rails App Server.
- `deployment`— Si ces attributs sont associés à un déploiement. Défini sur `null` pour cet exemple, car ils sont associés à un événement Configure.
- `valid_client_activities`— Une liste des activités valides des clients.

L'attribut `opsworks` est suivi de plusieurs autres attributs de niveau supérieur, y compris les attributs suivants :

- `opsworks_custom_cookbooks`— Si les livres de recettes personnalisés sont activés. Si tel est le cas, l'attribut inclut une liste de recettes personnalisées.
- `recipes`— Les recettes élaborées dans le cadre de cette activité.
- `opsworks_rubygems`— RubyGems Version de l'instance.
- `ssh_users`— Une liste d'utilisateurs SSH ; aucun dans cet exemple.
- `opsworks_bundler`— La version du bundler et son activation.
- `deploy`— Informations sur les activités de déploiement ; aucune dans cet exemple.

Attributs de déploiement

Les attributs d'un événement Deploy ou d'une commande de pile Execute Recipes se composent des attributs intégrés de configuration et de déploiement de pile, et des attributs personnalisés de pile ou de déploiement (aucun pour cet exemple). L'objet JSON suivant montre les attributs de php-app1 qui sont associés à un événement Deploy ayant déployé l'application SimplePHP sur les instances PHP de la pile. Une grande partie de l'objet se compose d'attributs de configuration de pile, similaires à ceux de l'événement Configure décrit dans la section précédente ; par conséquent, l'exemple se concentre principalement sur les attributs spécifiques au déploiement. Pour une description détaillée des différents attributs, consultez Attributs de déploiement et de configuration de pile : Linux et Attributs des livres de recettes intégrés.

```
{
   ...
  "opsworks": {
    ...
    "activity": "deploy",
    "applications": [
      {
        "slug_name": "simplephp",
        "name": "SimplePHP",
        "application_type": "php"
      }
    ],
    "deployment": "5e6242d7-8111-40ee-bddb-00de064ab18f",
    ...
  },
  ...
{
  "ssh_users": {
  },
  "deploy": {
    "simplephpapp": {
      "application": "simplephpapp",
      "application_type": "php",
      "environment_variables": {
        "USER_ID": "168424",
        "USER_KEY": "somepassword"
      },
      "auto_bundle_on_deploy": true,
      "deploy_to": "/srv/www/simplephpapp",
```

```
      "deploying_user": "arn:aws:iam::123456789012:user/guysm",
      "document_root": null,
      "domains": [
        "simplephpapp"
      ],
      "migrate": false,
      "mounted_at": null,
      "rails_env": null,
      "restart_command": "echo 'restarting app'",
      "sleep_before_restart": 0,
      "ssl_support": false,
      "ssl_certificate": null,
      "ssl_certificate_key": null,
      "ssl_certificate_ca": null,
      "scm": {
        "scm_type": "git",
        "repository": "git://github.com/amazonwebservices/opsworks-demo-php-simple-
app.git",
        "revision": "version1",
        "ssh_key": null,
        "user": null,
        "password": null
      },
      "symlink_before_migrate": {
        "config/opsworks.php": "opsworks.php"
      },
      "symlinks": {
      },
      "database": {
      },
      "memcached": {
        "host": null,
        "port": 11211
      },
      "stack": {
        "needs_reload": false
      }
    }
  },
}
```

L'attribut `opsworks` est largement identique à l'exemple de la section précédente. Les sections suivantes sont plus pertinentes pour le déploiement :

- `activity`— L'événement associé à ces attributs ; un événement Deploy dans cet exemple.
- `applications`— Contient un ensemble d'attributs pour chaque application qui fournissent les noms, les noms des slugs et les types des applications.

  Le nom du slug est le nom abrégé de l'application, que AWS OpsWorks Stacks génère à partir du nom de l'application. Le nom slug de SimplePHP est simplephp.
- `deployment`— L'ID de déploiement, qui identifie de manière unique un déploiement.

L'attribut `deploy` inclut des informations sur les applications qui sont déployées. Par exemple, les recettes intégrées Deploy utilisent les données de l'attribut `deploy` pour installer les fichiers dans les répertoires appropriés et créer les fichiers de connexion de base de données. L'attribut `deploy` inclut un attribut pour chaque application déployée, nommé d'après le nom court de l'application. Chaque attribut d'application contient les attributs suivants :

- `environment_variables`— Contient toutes les variables d'environnement que vous avez définies pour l'application. Pour de plus amples informations, veuillez consulter Variables d'environnement.
- `domains`— Par défaut, le domaine est le nom abrégé de l'application, qui est simplephpapp dans cet exemple. Si vous avez attribué des domaines personnalisés, ils apparaissent ici aussi bien. Pour de plus amples informations, veuillez consulter Utilisation des domaines personnalisés.
- `application`— Le nom abrégé de l'application.
- `scm`— Cet élément contient les informations requises pour télécharger les fichiers de l'application depuis son dépôt, un dépôt Git dans cet exemple.
- `database`— Informations de base de données, si la pile inclut une couche de base de données.
- `document_root`— La racine du document, qui est définie sur `null` dans cet exemple, indique que la racine est publique.
- `ssl_certificate_ca`,`ssl_support`, `ssl_certificate_key` — Indique si l'application prend en charge le protocole SSL. Si oui, les attributs `ssl_certificate_key` et `ssl_certificate_ca` sont définis avec les certificats correspondants.
- `deploy_to`— Le répertoire racine de l'application.

Les bases des livres de recettes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une pile AWS OpsWorks Stacks au niveau de la production nécessite généralement une certaine personnalisation, ce qui implique souvent la mise en œuvre d'un livre de recettes Chef personnalisé avec une ou plusieurs recettes, fichiers d'attributs ou fichiers modèles. Cette rubrique est un didacticiel d'introduction à la mise en œuvre de livres de recettes pour AWS OpsWorks Stacks.

Pour plus d'informations sur la façon dont AWS OpsWorks Stacks utilise les livres de recettes, y compris une brève introduction générale aux livres de recettes, voir. Livres de recettes et recettes Pour plus d'informations sur la mise en œuvre et le test des recettes Chef, consultez Infrastructure menée par les tests avec Chef, 2e édition.

Les exemples de ce didacticiel sont divisés en deux parties :

- Principes de base des livre de recettes est un ensemble d'exemples de procédures destinés aux utilisateurs qui ne connaissent pas Chef ; les utilisateurs expérimentés de Chef peuvent ignorer cette section.

  Les exemples présentent les bases de la mise en œuvre des livres de recettes pour effectuer des tâches courantes, telles que l'installation de paquets ou la création de répertoires. Pour simplifier le processus, vous allez utiliser deux outils intéressants, Vagrant et Test Kitchen, afin d'exécuter la plupart des exemples localement sur une machine virtuelle. Avant de commencer Principes de base des livre de recettes, vous devez lire Vagrant et Test Kitchen afin d'apprendre à installer et à utiliser ces outils. Dans la mesure où Test Kitchen ne prend pas encore en charge Windows, les exemples sont tous destinés à Linux, avec des notes indiquant comment les adapter à Windows.
- Implémentation de Cookbooks for Stacks AWS OpsWorksdécrit comment implémenter des recettes pour les AWS OpsWorks piles, y compris pour les piles Windows.

  Il inclut également des informations plus avancées, telles que la façon d'utiliser Berkshelf pour gérer des livres de cuisine externes. Les exemples sont destinés aux nouveaux utilisateurs

de Chef, en grande partie comme les exemples de Principes de base des livre de recettes. Cependant, AWS OpsWorks Stacks fonctionne un peu différemment du serveur Chef. Nous recommandons donc aux utilisateurs expérimentés de Chef de lire au moins cette section.

Vagrant et Test Kitchen

Si vous utilisez des recettes pour les instances Linux, Vagrant et Test Kitchen sont des outils très utiles pour l'apprentissage, ainsi que le développement et les tests initiaux. Vous y trouverez de brèves descriptions de Vagrant et Test Kitchen, ainsi que des instructions d'installation et des procédures pas à pas qui vous permettront de configurer et de vous familiariser avec les bases de l'utilisation des outils. Bien que Vagrant prenne en charge Windows, ce n'est pas le cas de Test Kitchen, c'est pourquoi seuls des exemples avec Linux sont proposés pour ces outils.

Vagrant

Vagrant fournit un environnement uniforme pour l'exécution et le test du code sur une machine virtuelle. Il prend en charge une grande variété d'environnements, appelés boîtes Vagrant, chacun représentant un système d'exploitation configuré. Pour AWS OpsWorks Stacks, les environnements d'intérêt sont basés sur les distributions Ubuntu, Amazon ou Red Hat Enterprise Linux (RHEL). Les exemples utilisent donc principalement une boîte Vagrant nommée. `opscode-ubuntu-12.04`

Vagrant est disponible pour les systèmes Linux, Windows et Macintosh, c'est pourquoi vous pouvez utiliser votre poste de travail préféré pour implémenter et tester des recettes sur n'importe quel système d'exploitation pris en charge. Les exemples de ce chapitre ont été créés sur un système Ubuntu Linux, mais la traduction des procédures sur les systèmes Windows ou Macintosh est simple.

Vagrant est globalement une enveloppe pour un fournisseur de virtualisation. La plupart des exemples utilisent le VirtualBoxfournisseur. VirtualBox est gratuit et disponible pour les systèmes Linux, Windows et Macintosh. La procédure pas à pas de Vagrant fournit des instructions d'installation si vous ne les avez pas déjà VirtualBox sur votre système. Notez que vous pouvez exécuter des environnements basés sur Ubuntu, VirtualBox mais Amazon Linux n'est disponible que pour les instances Amazon. EC2 Cependant, vous pouvez exécuter un système d'exploitation similaire tel que CentOS VirtualBox, ce qui est utile pour le développement initial et les tests.

Pour plus d'informations sur d'autres fournisseurs, consultez la documentation sur Vagrant. Le fournisseur de `vagrant-aws` plug-in vous permet notamment d'utiliser Vagrant avec des EC2 instances Amazon. Ce fournisseur est particulièrement utile pour tester des recettes sur Amazon

Linux, qui n'est disponible que sur les EC2 instances Amazon. Le fournisseur `vagrant-aws` est gratuit, mais vous devez posséder un compte AWS et payer toutes les ressources AWS que vous utilisez.

À ce stade, vous devez parcourir la procédure pas à pas de mise en route de Vagrant, qui décrit l'installation de Vagrant sur votre poste de travail et vous enseigne les bases de son utilisation. Veuillez noter que les exemples de ce chapitre n'utilisent pas de référentiel Git ; vous pouvez donc ignorer cette partie de la procédure si vous le souhaitez.

Test Kitchen

Test Kitchen simplifie le processus d'exécution et de test de vos livres de recettes sur Vagrant. Sur le plan pratique, vous n'utilisez presque jamais Vagrant directement. Test Kitchen effectue les tâches les plus courantes, notamment :

- Lancement d'une instance dans Vagrant.
- Transfert de livres de recettes vers l'instance.
- Exécution des recettes du livre de recettes sur l'instance.
- Test de recettes d'un livre de recettes sur l'instance.
- Utilisation de SSH pour se connecter à l'instance.

Au lieu d'installer Test Kitchen directement, nous recommandons d'installer Chef DK. En plus de Chef lui-même, ce package comprend Test Kitchen, Berkshelf et plusieurs autres outils utiles. ChefSpec

À ce stade, vous devez passer à la Procédure pas à pas de mise en route de Test Kitchen, qui vous enseigne les bases de l'utilisation de Test Kitchen pour exécuter et tester des recettes.

> **Note**
>
> Les exemples de ce chapitre utilisent Test Kitchen pour exécuter facilement des recettes. Si vous préférez, vous pouvez arrêter la procédure de mise en route à la fin de la section Vérification manuelle, qui couvre tout ce que vous avez besoin de connaître pour les exemples. Cependant, Test Kitchen est essentiellement une plateforme de test qui prend en charge les infrastructures de test, par exemple bash automated test system (BATS). Vous devrez terminer le reste de la procédure pas à pas à un moment donné afin d'apprendre à utiliser Test Kitchen pour tester vos recettes.

Principes de base des livre de recettes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez utiliser les livres de recettes pour exécuter une grande variété de tâches. Les rubriques suivantes supposent que vous débutez avec Chef et expliquent comment utiliser les livres de recettes afin de réaliser certaines tâches courantes. Dans la mesure où Test Kitchen ne prend pas encore en charge Windows, les exemples sont tous destinés à Linux, avec des notes indiquant comment les adapter à Windows. Si vous découvrez Chef, nous vous recommandons de parcourir ces exemples, même si vous travaillez avec Windows. La plupart des exemples de cette rubrique peuvent être utilisés sur les instances Windows avec quelques légères modifications, qui sont indiquées dans les exemples. Tous les exemples s'exécutent sur une machine virtuelle, donc vous n'avez pas besoin d'avoir un ordinateur Linux. Il suffit d'installer Vagrant et Test Kitchen sur votre poste de travail habituel.

> **Note**
>
> Si vous souhaitez exécuter ces recettes sur une instance Windows, l'approche la plus simple consiste à créer une pile Windows et à exécuter les recettes sur l'une des instances de la pile. Pour plus d'informations sur la façon d'exécuter des recettes sur une instance AWS OpsWorks Stacks Windows, consultezExécution d'une recette sur une instance Windows.

Avant de poursuivre, assurez-vous que vous avez installé Vagrant et Test Kitchen et que vous avez suivi les procédures de mise en route. Pour de plus amples informations, veuillez consulter Vagrant et Test Kitchen.

Rubriques

- Structure de la recette
- Exemple 1 : Installation des packages

- Exemple 2 : Gestion des utilisateurs
- Exemple 3 : Création de répertoires
- Exemple 4 : Ajout du contrôle de flux
- Exemple 5 : Utilisation d'attributs
- Exemple 6 : Création de fichiers
- Exemple 7 : Exécution des commandes et des scripts
- Exemple 8 : Gestion des services
- Exemple 9 : utilisation des EC2 instances Amazon
- Étapes suivantes

Structure de la recette

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Un livre de recettes est principalement constitué d'un ensemble de recettes qui peuvent effectuer diverses tâches sur une instance. Pour clarifier l'implémentation des recettes, il est utile d'examiner un exemple simple. Voici la recette de configuration de la couche HAProxy intégrée. Concentrez-vous simplement sur la structure globale à ce stade et ne vous inquiétez pas trop des détails ; ils seront couverts dans les exemples suivants.

```
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**
>
> Pour cet exemple et pour d'autres exemples de recettes fonctionnelles et de fichiers associés, consultez les recettes intégrées d'AWS OpsWorks Stacks.

L'exemple met en évidence les éléments principaux de la recette, qui sont décrits dans les sections suivantes.

Rubriques

- Ressources
- Contrôle de flux
- Recettes incluses

Ressources

Les recettes se composent en grande partie d'un ensemble de ressources Chef. Chacun spécifie un aspect particulier de l'état final de l'instance, par exemple un package à installer ou un service à démarrer. L'exemple a quatre ressources :

- Une `package` ressource, qui représente un package installé, un HAProxy serveur pour cet exemple.

- Une `service` ressource, qui représente un service, le HAProxy service dans cet exemple.
- Deux `template` ressources, qui représentent des fichiers à créer à partir d'un modèle spécifié, deux fichiers HAProxy de configuration pour cet exemple.

Les ressources permettent de spécifier l'état de l'instance de façon déclarative. En arrière-plan, chaque ressource a un fournisseur associé qui effectue les tâches requises, telles que l'installation des packages, la création et la configuration des répertoires, le démarrage des services, etc. Si les détails de la tâche dépendent du système d'exploitation, la ressource a plusieurs fournisseurs et utilise celui qui est approprié pour le système. Par exemple, sur un système Red Hat Linux, le fournisseur `package` utilise `yum` pour installer les packages. Sur un système Ubuntu Linux, le fournisseur `package` utilise `apt-get`.

Vous implémentez une ressource comme un bloc de code Ruby avec le format général suivant.

```
resource_type "resource_name" do
  attribute1 'value1'
  attribute2 'value2'
  ...
  action :action_name
  notifies : action 'resource'
end
```

Les éléments sont les suivants :

Type de ressource

(Obligatoire) L'exemple inclut trois types de ressource, `package`, `service` et `template`.

Nom de la ressource

(Obligatoire) Le nom identifie la ressource particulière et est parfois utilisé comme valeur par défaut pour l'un des attributs. Dans l'exemple, `package` représente une ressource de package nommée `haproxy` et la première ressource `template` représente un fichier de configuration nommé `/etc/default/haproxy`.

Attributs

(Facultatif) les attributs spécifient la configuration des ressources et varient en fonction du type de ressource et de la façon dont vous souhaitez configurer la ressource.

- Les ressources `template` de l'exemple définissent explicitement un ensemble d'attributs qui spécifient la source, le propriétaire, le groupe et le mode du fichier créé.
- Les ressources `package` et `service` de l'exemple ne définissent pas explicitement les attributs.

  Le nom de la ressource est généralement la valeur par défaut pour un attribut obligatoire et il arrive que rien d'autre ne soit nécessaire. Par exemple, le nom de la ressource est la valeur par défaut pour la ressource `package` dont le seul attribut obligatoire est `package_name`.

Il existe également des attributs spécialisés nommés attributs de protection, qui spécifient lorsque le fournisseur de la ressource va intervenir. Par exemple, l'attribut `only_if` indique au fournisseur de ressources d'agir uniquement si une condition spécifiée est remplie. La HAProxy recette n'utilise pas d'attributs de garde, mais ils sont utilisés dans plusieurs des exemples suivants.

Actions et notifications

(Facultatif) Les actions et les notifications spécifient les tâches qui seront effectuées par le fournisseur.

- `action` indique au fournisseur d'effectuer une action spécifiée, telle que l'installation ou la création.

  Chaque ressource dispose d'un ensemble d'actions qui dépendent de la ressource particulière, dont l'action par défaut. Dans l'exemple, l'action de la ressource `package` action de la ressource est `install`, qui demande au fournisseur d'installer le package. La première ressource `template` n'a pas d'élément `action`, c'est pourquoi le fournisseur prend l'action `create` par défaut.
- `notifies` indique au fournisseur de la ressource d'effectuer une action, mais uniquement si l'état de la ressource a été modifié.

  `notifies` est généralement utilisé avec des ressources telles que `template` et `file` pour exécuter des tâches telles que le redémarrage d'un service après la modification d'un fichier de configuration. Les ressources n'ont pas de notifications par défaut. Si vous souhaitez une notification, la ressource doit avoir un élément `notifies` explicite. Dans la HAProxy recette, la deuxième `template` ressource demande à la `service` ressource haproxy de redémarrer le HAProxy service si le fichier de configuration associé a changé.

Les ressources dépendent parfois du système d'exploitation.

- Certaines ressources peuvent être utilisées uniquement sur les systèmes Linux ou Windows.

  Par exemple, package installe les packages sur les systèmes Linux et windows_package installe les packages sur les systèmes Windows.
- Certaines ressources peuvent être utilisées avec n'importe quel système d'exploitation, mais ont des attributs qui sont propres à un système particulier.

  Par exemple, la ressource file peut être utilisée sur les systèmes Linux ou Windows, mais possède des ensembles distincts d'attributs pour configurer des autorisations.

Pour les descriptions de ressources standard, y compris les attributs, les actions et les notifications disponibles pour chaque ressource, consultez À propos des ressources et des fournisseurs.

Contrôle de flux

Dans la mesure où les recettes sont des applications Ruby, vous pouvez utiliser les structures de contrôle Ruby pour intégrer le contrôle de flux dans une recette. Par exemple, vous pouvez utiliser une logique conditionnelle Ruby pour que la recette se comporte différemment selon les systèmes. La HAProxy recette inclut un `if` bloc qui utilise une `template` ressource pour créer un fichier de configuration, mais uniquement si la recette est exécutée sur un système Debian ou Ubuntu.

Un autre scénario courant consiste à utiliser une boucle pour exécuter une ressource plusieurs fois avec différents paramètres d'attribut. Par exemple, vous pouvez créer un ensemble de répertoires en utilisant une boucle pour exécuter une ressource `directory` plusieurs fois avec différents noms de répertoires.

> **Note**
>
> Si vous ne maîtrisez pas bien Ruby, consultez la section Juste assez de Ruby pour Chef, qui couvre ce que vous devez savoir pour la plupart des recettes.

Recettes incluses

`include_recipe` inclut d'autres recettes dans votre code, ce qui vous permet d'organiser os recettes par modules et de réutiliser le même code dans plusieurs recettes. Lorsque vous exécutez la recette de l'hôte, Chef remplace chaque élément `include_recipe` par le code de la recette spécifiée avant d'exécuter la recette de l'hôte. Vous identifiez une recette incluse à l'aide de la syntaxe standard de Chef `cookbook_name::recipe_name`, avec `recipe_name`

qui omet l'extension `.rb`. L'exemple inclut une recette qui représente le HAProxy service. `haproxy::service`

> **Note**
>
> Si vous utilisez `include_recipe` dans les recettes qui exécutent Chef 11.10 et des versions ultérieures pour inclure une recette d'un autre livre de recettes, vous devez utiliser une instruction `depends` pour déclarer la dépendance dans le fichier `metadata.rb` du livre de recettes. Pour de plus amples informations, veuillez consulter Mise en œuvre des recettes : Chef 11.10.

Exemple 1 : Installation des packages

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

L'installation des packages est l'une des utilisations plus courantes des recettes et peut être très simple, selon le package. Par exemple, la recette suivante installe Git sur un système Linux.

```
package 'git' do
  action :install
end
```

La ressource `package` gère l'installation des packages. Pour cet exemple, vous n'avez pas besoin de spécifier d'attributs. Le nom de la ressource est la valeur par défaut pour l'attribut `package_name`, qui identifie le package. L'action `install` demande au fournisseur d'installer le package. Vous pouvez rendre le code encore plus simple en ignorant `install` ; c'est l'action par défaut de la ressource `package` action par défaut de la ressource. Lorsque vous exécutez la recette, le Chef utilise le fournisseur approprié pour installer le package. Sur le système Ubuntu que vous utiliserez pour l'exemple, le fournisseur installe Git en appelant `apt-get`.

> **Note**
>
> L'installation de logiciels sur un système Windows nécessite une procédure légèrement différente. Pour de plus amples informations, veuillez consulter Installation de logiciels Windows.

Pour utiliser Test Kitchen afin d'exécuter cette recette dans Vagrant, vous devez d'abord configurer un livre de recettes, puis initialiser et configurer Test Kitchen. La procédure suivante est destinée à un système Linux, mais elle est globalement similaire pour les systèmes Windows et Macintosh. Commencez par ouvrir une fenêtre de terminal ; tous les exemples dans ce chapitre utilisent les outils de ligne de commande.

Pour préparer le livre de recettes

1. Dans votre répertoire de base, créez un sous-répertoire nommé `opsworks_cookbooks` qui contiendra tous les livres de recettes de ce chapitre. Créez ensuite un sous-répertoire pour ce livre de recettes nommé `installpkg` et accédez à celui-ci.
2. Dans `installpkg`, créez un fichier nommé `metadata.rb` qui contient le code suivant.

   ```
   name "installpkg"
   version "0.1.0"
   ```

   Pour plus de simplicité, les exemples de ce chapitre spécifient simplement le nom et la version du livre de recettes, mais `metadata.rb` peut contenir diverses métadonnées de livre. Pour plus d'informations, consultez À propos des métadonnées des livres de recettes.

   > **Note**
   >
   > Veillez à créer `metadata.rb` avant d'initialiser Test Kitchen ; Il utilise les données pour créer le fichier de configuration par défaut.

3. Dans `installpkg`, exécutez `kitchen init`, qui initialise Test Kitchen et installe le pilote de Vagrant par défaut.
4. La commande `kitchen init` crée un fichier de configuration YAML dans `installpkg` nommé `.kitchen.yml`. Ouvrez le fichier dans votre éditeur de texte favori. Le fichier `.kitchen.yml` inclut une section `platforms` qui spécifie les systèmes sur lesquels vous

devez exécuter les recettes. Test Kitchen crée une instance et exécute les recettes spécifiés sur chaque plateforme.

> **Note**
>
> Par défaut, Test Kitchen exécute les recettes sur une plateforme à la fois. Si vous ajoutez un argument `-p` à n'importe quelle commande qui crée une instance, Test Kitchen exécutera les recettes sur toutes les plateformes, en parallèle.

Une seule plateforme est suffisante pour cet exemple, vous devez donc modifier `.kitchen.yml` de façon à supprimer la plateforme `centos-6.4`. La structure de votre fichier `.kitchen.yml` devrait maintenant ressembler à ceci :

```
---
driver:
  name: vagrant

provisioner:
  name: chef_solo

platforms:
  - name: ubuntu-12.04

suites:
  - name: default
    run_list:
      - recipe[installpkg::default]
    attributes:
```

Test Kitchen exécute uniquement les recettes qui sont dans la liste d'exécution de `.kitchen.yml`. Vous identifiez les recettes en utilisant le `[`*`cookbook_name`*`::`*`recipe_name`*`]` format, où l'`.rb`extension est *`recipe_name`* omise. Au début, la liste d'exécution de `.kitchen.yml` contient la recette par défaut du livre de recettes, `installpkg::default`. C'est la recette que vous allez implémenter, c'est pourquoi vous n'avez pas besoin de modifier la liste d'exécution.

5. Créez un sous-répertoire de `installpkg` nommé `recipes`.

Si un livre de recettes contient des recettes, comme la plupart d'entre elles, elles doivent se trouver dans le sous-répertoire. `recipes`

Vous pouvez désormais ajouter la recette au livre de recettes et utiliser Test Kitchen pour l'exécuter sur une instance.

Pour exécuter la recette

1. Créez un fichier nommé `default.rb` qui contient le code d'exemple de l'installation de Git dès le début de la section et l'enregistrer dans le sous-répertoire `recipes`.
2. Dans le répertoire `installpkg`, exécutez `kitchen converge`. Cette commande démarre une nouvelle instance Ubuntu dans Vagrant, copie vos livres de recettes sur l'instance et lance une exécution Chef pour exécuter les recettes de la liste d'exécutions. `.kitchen.yml`
3. Pour vérifier que la recette a réussi, exécutez `kitchen login`, qui ouvre une connexion SSH à l'instance. Exécutez ensuite `git --version` pour vérifier que Git a été installé correctement. Pour revenir à votre poste de travail, exécutez `exit`.
4. Lorsque vous avez terminé, exécutez `kitchen destroy` pour arrêter l'instance. L'exemple suivant utilise un autre livre de recettes.

Cet exemple était un bon moyen de démarrer, mais il est particulièrement simple. D'autres packages peuvent être plus complexes à installer ; vous devrez peut-être faire tout ou partie des éléments suivants :

- Créez et configurez un utilisateur.
- Créez un ou plusieurs répertoires pour les données, les journaux, etc.
- Installez un ou plusieurs fichiers de configuration.
- Spécifiez des valeurs de nom ou d'attribut de package différentes selon les systèmes d'exploitation.
- Démarrez un service, puis redémarrez-le en fonction des besoins.

Les exemples suivants expliquent comment résoudre ces problèmes, ainsi que d'autres opérations utiles.

Exemple 2 : Gestion des utilisateurs

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une autre tâche simple consiste à gérer les utilisateurs sur une instance. La recette suivante ajoute un utilisateur à une instance Linux.

```
user "myuser" do
  home "/home/newuser"
  shell "/bin/bash"
end
```

Vous utilisez une ressource user pour gérer les utilisateurs sur les systèmes Linux et Windows, bien que certains attributs s'appliquent à un seul système. L'exemple crée un utilisateur nommé `myuser` et spécifie son répertoire de base et shell. Il n'y a aucune action spécifiée, c'est pourquoi la ressource utilise l'action `create` par défaut. Vous pouvez ajouter des attributs à `user` pour spécifier divers autres paramètres, tels que leur mot de passe ou ID de groupe. Vous pouvez également utiliser `user` pour les tâches de gestion des utilisateurs connexes, telles que la modification des paramètres de l'utilisateur ou la suppression d'utilisateurs. Pour plus d'informations, consultez user.

Pour exécuter la recette

1. Créez un répertoire dans `opsworks_cookbooks`, nommé `newuser` et accédez à celui-ci.
2. Créez un fichier `metadata.rb` qui contient le code suivant et enregistrez-le dans `newuser`.

   ```
   name "newuser"
   version "0.1.0"
   ```

3. Initialisez et configurez Test Kitchen, comme décrit dans Exemple 1 : Installation des packages et ajoutez un répertoire `recipes` dans le répertoire `newuser`.

4. Ajoutez un fichier `default.rb` avec l'exemple de recette dans le répertoire `recipes` du livre de recettes.
5. Exécutez `kitchen converge` pour exécuter la recette.
6. Utilisez `kitchen login` pour vous connecter à l'instance et vérifier l'existence du nouvel utilisateur en exécutant `cat /etc/passwd`. L'utilisateur `myuser` doit être en bas du fichier.

Exemple 3 : Création de répertoires

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Lorsque vous installez un package sur une instance, vous devez souvent créer des fichiers de configuration et les placer dans les répertoires appropriés. Toutefois, ces répertoires peuvent ne pas exister à ce stade. De plus, vous devrez parfois créer des répertoires pour les données, les fichiers journaux, etc. Par exemple, vous démarrez d'abord le système Ubuntu que vous utilisez pour la plupart des exemples, le `/srv` répertoire ne contient aucun sous-répertoire. Si vous installez un serveur d'applications, vous aurez probablement besoin d'un répertoire `/srv/www/`, voire de quelques sous-répertoires pour les fichiers de données, les journaux, etc. La recette suivante crée `/srv/www/` sur une instance.

```
directory "/srv/www/" do
  mode 0755
  owner 'root'
  group 'root'
  action :create
end
```

Vous utilisez une ressource `directory` pour créer et configurer des répertoires sur les systèmes Linux et Windows, bien que certains attributs soient utilisés différemment. Le nom de la ressource est la valeur par défaut de l'attribut `path` de la ressource. L'exemple crée donc `/srv/www/` et spécifie ses propriétés `mode`, `owner` et `group`.

Pour exécuter la recette

1. Créez un répertoire dans `opsworks_cookbooks`, nommé `createdir` et accédez à celui-ci.
2. Initialisez et configurez Test Kitchen, comme décrit dans Exemple 1 : Installation des packages et ajoutez un répertoire `recipes` dans `createdir`.
3. Ajoutez un fichier `default.rb` avec le code de recette dans le sous-répertoire `recipes` du livre de recettes.
4. Exécutez `kitchen converge` pour exécuter la recette.
5. Exécutez `kitchen login`, accédez à `/srv` et vérifiez qu'il possède un sous-répertoire `www`.
6. Exécutez `exit` pour revenir à votre poste de travail, mais n'interrompez pas l'exécution de l'instance.

> **Note**
>
> Pour créer un répertoire relatif à votre répertoire de base sur l'instance, utilisez `#{ENV['HOME']}` pour représenter le répertoire de base. Par exemple, ce qui suit crée le répertoire `~/shared`.
>
> ```
> directory "#{ENV['HOME']}/shared" do
>   ...
> end
> ```

Supposons que vous souhaitiez créer un répertoire plus imbriqué, par exemple `/srv/www/shared`. Vous pouvez modifier la recette précédente comme suit.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  action :create
end
```

Pour exécuter la recette

1. Remplacez le code de `default.rb` par la recette précédente.
2. Exécutez `kitchen converge` à partir du répertoire `createdir`.
3. Pour vérifier que le répertoire a bien été créé, exécutez `kitchen login`, accédez à `/srv/www` et vérifiez qu'il contient un sous-répertoire `shared`.
4. Exécutez `kitchen destroy` pour arrêter l'instance.

Vous remarquerez que la commande `kitchen converge` a été exécutée nettement plus rapidement. En effet, l'instance est déjà en cours d'exécution et il n'est donc pas nécessaire de la démarrer, d'installer Chef, etc. Test Kitchen se contente de copier le livre de recettes mis à jour sur l'instance et démarre une exécution de Chef.

Exécutez maintenant `kitchen converge` à nouveau afin d'exécuter la recette sur une nouvelle instance. Vous voyez maintenant le résultat suivant.

```
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]
>>>>>> ----------------------
```

Que s'est-il passé ? Le problème est que, par défaut, une ressource `directory` peut créer un seul répertoire à la fois ; elle ne peut pas créer de chaîne de répertoires. La raison pour laquelle la recette a fonctionné plus tôt est que la première recette que vous avez exécutée sur l'instance a déjà créé `/srv/www`, c'est pourquoi la création de `/srv/www/shared` a créé un seul sous-répertoire.

> **Note**
>
> Lorsque vous exécutez `kitchen converge`, veillez à savoir si vous exécutez vos recettes sur une nouvelle instance ou sur une instance existante. Vous obtiendrez peut-être des résultats différents.

Pour créer une chaîne de sous-répertoires, ajoutez un attribut `recursive` à `directory` et définissez-le sur `true`. La recette suivante crée `/srv/www/shared` directement sur une instance propre.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end
```

Exemple 4 : Ajout du contrôle de flux

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Certaines recettes sont une simple série de ressources Chef. Dans ce cas, lorsque vous exécutez la recette, elle exécute simplement chaque fournisseur de ressources dans l'ordre. Cependant, il est souvent utile d'avoir un chemin d'exécution plus sophistiqué. Voici deux scénarios courants :

- Vous voulez qu'une recette exécute la même ressource plusieurs fois avec des paramètres d'attribut différents.
- Vous souhaitez utiliser différents paramètres d'attribut sur différents systèmes d'exploitation.

Vous pouvez traiter ce type de scénario en intégrant les structures de contrôle Ruby à la recette. Cette section montre comment modifier la recette de Exemple 3 : Création de répertoires pour traiter les deux scénarios.

Rubriques

- Itération
- Logique conditionnelle

Itération

Exemple 3 : Création de répertoires a montré comment utiliser une ressource `directory` pour créer un répertoire ou une chaîne de répertoires. Toutefois, supposons que vous souhaitiez créer deux répertoires distincts, `/srv/www/config` et `/srv/www/shared`. Vous pouvez implémenter une ressource de répertoire distincte pour chaque répertoire, mais cette approche peut être fastidieuse si vous souhaitez créer un très grand nombre de répertoires. La recette suivante montre un moyen plus simple de gérer la tâche.

```
[ "/srv/www/config", "/srv/www/shared" ].each do |path|
  directory path do
    mode 0755
    owner 'root'
    group 'root'
    recursive true
    action :create
  end
end
```

Au lieu d'utiliser une ressource de répertoire distincte pour chaque sous-répertoire, la recette utilise un ensemble de chaînes qui contient les chemins de sous-répertoire. La méthode `each` de Ruby exécute la ressource une fois pour chaque élément de l'ensemble, en commençant par le premier. La valeur de l'élément est représentée dans la ressource par la variable `path`, qui représente dans ce cas le chemin d'accès au répertoire. Vous pouvez facilement adapter cet exemple afin de créer n'importe quel nombre de sous-répertoires.

Pour exécuter la recette

1. Restez dans le répertoire `createdir`. Vous allez utiliser ce livre de recettes pour les prochains exemples.

2. Si vous ne l'avez pas déjà fait, exécutez `kitchen destroy` afin de commencer avec une instance propre.
3. Remplacez le code de `default.rb` par l'exemple et exécutez `kitchen converge`.
4. Connectez-vous à l'instance. Vous verrez les répertoires nouvellement créés sous `/srv`.

Vous pouvez utiliser une table de hachage pour spécifier deux valeurs pour chaque itération. La recette suivante crée `/srv/www/config` et `/srv/www/shared`, chacun avec un mode différent.

```
{ "/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
```

Pour exécuter la recette

1. Si vous ne l'avez pas déjà fait, exécutez `kitchen destroy` afin de commencer avec une instance propre.
2. Remplacez le code de `default.rb` par l'exemple et exécutez `kitchen converge`.
3. Connectez-vous à l'instance. Vous verrez les répertoires nouvellement créés sous `/srv` avec les modes spécifiés.

> **Note**
>
> AWS OpsWorks Les recettes Stacks utilisent généralement cette approche pour extraire des valeurs du JSON de configuration et de déploiement de la pile (qui est essentiellement une grande table de hachage) et les insérer dans une ressource. Pour obtenir un exemple, consultez Recettes Deploy.

Logique conditionnelle

Vous pouvez également utiliser la logique conditionnelle de Ruby pour créer plusieurs branches d'exécution. La recette suivante utilise la logique `if-elsif-else` afin d'étendre l'exemple précédent pour qu'il crée un sous-répertoire nommé `/srv/www/shared`, mais uniquement sur les systèmes Debian et Ubuntu. Pour tous les autres systèmes, il consigne un message d'erreur qui s'affiche dans la sortie de Test Kitchen.

```
if platform?("debian", "ubuntu")
  directory "/srv/www/shared" do
    mode 0755
    owner 'root'
    group 'root'
    recursive true
    action :create
  end
else
  log "Unsupported system"
end
```

Pour exécuter un exemple de recette

1. Si votre instance est toujours fonctionnelle, exécutez `kitchen destroy` pour la fermer.
2. Remplacez le code de `default.rb` par l'exemple de code.
3. Modifiez `.kitchen.yml` afin d'ajouter un système CentOS 6.4 à la liste de plateformes. La section `platforms` du fichier doit maintenant ressembler à ce qui suit.

   ```
   ...
   platforms:
     - name: ubuntu-12.04
     - name: centos-6.4
   ...
   ```

4. Exécutez `kitchen converge`, qui crée une instance et exécute les recettes pour chaque plateforme dans `.kitchen.yml`, dans l'ordre.

> **Note**
>
> Si vous souhaitez faire converger une seule instance, ajoutez le nom de l'instance comme paramètre. Par exemple, pour faire converger la recette uniquement sur la plateforme Ubuntu, exécutez `kitchen converge default-ubuntu-1204`. Si vous oubliez les noms de plateformes, il suffit d'exécuter `kitchen list`.

Vous devriez voir votre message de journal dans la partie CentOS de la sortie Test Kitchen, qui ressemblera à quelque chose de similaire à ce qui suit :

```
...
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
```

Vous pouvez désormais vous connecter aux instances et vérifier que les répertoires ont été créés ou non. Toutefois, vous ne pouvez pas simplement exécuter `kitchen login` maintenant. Vous devez spécifier l'instance en ajoutant le nom de la plateforme, par exemple `kitchen login default-ubuntu-1204`.

> **Note**
>
> Si une commande de Test Kitchen prend un nom d'instance, vous n'avez pas besoin de taper le nom complet. Test Kitchen traite le nom d'instance comme une expression régulière Ruby, vous avez donc besoin de suffisamment de caractères pour fournir une correspondance unique et de rien de plus. Par exemple, vous pouvez converger uniquement l'instance Ubuntu en exécutant `kitchen converge ub` ou vous connecter à l'instance CentOS en exécutant `kitchen login 64`.

A ce stade, vous vous demandez probablement comment la recette sait sur quelle plateforme elle est exécutée. Chef exécute un outil appelé Ohai pour chaque exécution qui collecte les données

du système, y compris la plateforme, et il la représente sous forme d'un ensemble d'attributs dans une structure nommé objet de nœud. La méthode `platform?` de Chef compare les systèmes entre parenthèses à la valeur de plateforme Ohai et retourne true en cas de correspondance.

Vous pouvez référencer la valeur d'un attribut de nœud directement dans votre code en utilisant `node['attribute_name']`. La valeur de la plateforme, par exemple, est représentée par `node['platform']`. Vous pouvez, par exemple, avoir écrit l'exemple précédent comme suit.

```
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
```

Les utilisateurs incluent souvent une logique conditionnelle dans une recette pour tenir compte du fait que les différentes familles Linux utilisent parfois des noms différents pour les packages, les répertoires, etc. Par exemple, le nom du package Apache est `httpd` sur les systèmes CentOS et `apache2` sur les systèmes Ubuntu.

Si vous avez juste besoin d'une chaîne différente pour différents systèmes, la méthode `value_for_platform` de Chef est une solution plus simple que `if-elsif-else`. La recette suivante crée un répertoire `/srv/www/shared` sur les systèmes CentOS, un répertoire `/srv/www/data` sur les systèmes Ubuntu et `/srv/www/config` sur tous les autres.

```
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` attribue le chemin d'accès approprié pour `data_dir` et la ressource `directory` utilise cette valeur pour créer le répertoire.

Pour exécuter un exemple de recette

1. Si votre instance est toujours fonctionnelle, exécutez `kitchen destroy` pour la fermer.
2. Remplacez le code de `default.rb` par l'exemple de code.
3. Exécutez `kitchen converge`, puis connectez-vous à chaque instance pour vérifier que les répertoires appropriés sont présents.

Exemple 5 : Utilisation d'attributs

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les recettes des sections précédentes ont utilisé des valeurs codées en dur pour tout ce qui n'était pas la plateforme. Cette approche peut être gênante si, par exemple, vous souhaitez utiliser la même valeur dans plusieurs recettes. Vous pouvez définir des valeurs séparément à partir des recettes en incluant un fichier d'attribut dans votre livre de recettes.

Un fichier d'attribut est une application Ruby qui attribue des valeurs à un ou plusieurs attributs. Il doit se trouver dans le dossier `attributes` du livre de recettes. Chef intègre les attributs dans l'objet de nœud et n'importe quelle recette peut utiliser les valeurs d'attribut en référençant ce dernier. Cette rubrique montre comment modifier la recette de Itération afin d'utiliser les attributs. Voici la recette d'origine à titre de référence.

```
[ "/srv/www/config", "/srv/www/shared" ].each do |path|
  directory path do
```

```
    mode 0755
    owner 'root'
    group 'root'
    recursive true
    action :create
  end
end
```

Ce qui suit définit les attributs pour le nom du sous-répertoire, le mode, le propriétaire et les valeurs de groupe.

```
default['createdir']['shared_dir'] = 'shared'
default['createdir']['config_dir'] = 'config'
default['createdir']['mode'] = 0755
default['createdir']['owner'] = 'root'
default['createdir']['group'] = 'root'
```

Remarques :

- Chaque définition commence par un type d'attribut.

  Si un attribut est défini plusieurs fois, par exemple dans différents fichiers d'attributs, le type d'attribut indique la priorité de l'attribut, qui détermine la définition incorporée dans l'objet nœud. Pour de plus amples informations, veuillez consulter Priorité des attributs. Toutes les définitions de cet exemple ont le type d'attribut `default`, qui est le type habituel pour ces cas de figure.
- Les attributs ont des noms imbriqués.

  L'objet de nœud est en fait une table de hachage qui peut s'imbriquer profondément de façon arbitraire, afin que les noms d'attributs puissent être imbriqués et le soient souvent. Ce fichier d'attribut suit une pratique standard d'utilisation d'un nom imbriqué avec le nom du livre de recettes, `createdir`, comme premier élément.

Vous utiliserez généralement createdir comme premier élément de l'attribut car, lors de l'exécution de Chef, ce dernier intègre ainsi les attributs de chaque livre de recettes dans l'objet de nœud. Avec AWS OpsWorks Stacks, l'objet node inclut un grand nombre d'attributs issus des livres de recettes intégrés, en plus des attributs que vous définissez. Le fait d'inclure le nom du livre de recettes dans le nom d'attribut réduit le risque de conflits de noms avec les attributs d'un autre livre de recettes, surtout si votre attribut a un nom tel que `port` ou `user`. Ne nommez pas un attribut de la façon

suivante `[:apache2][:user]`, par exemple, sauf si vous voulez remplacer cette valeur d'attribut. Pour de plus amples informations, veuillez consulter Utilisation d'attributs personnalisés de livres de recettes.

L'exemple suivant montre la recette originale qui utilise des attributs au lieu des valeurs codées en dur.

```
[ "/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**
>
> Si vous souhaitez intégrer une valeur d'attribut dans une chaîne, encapsulez-la avec `#{}`. Dans l'exemple précédent, `#{node['createdir']['shared_dir']}` ajoute « shared » à « /srv/www/ ».

Pour exécuter la recette

1. Exécutez `kitchen destroy` pour commencer avec une instance propre.
2. Remplacez le code de `recipes/default.rb` par l'exemple de recette précédent.
3. Créez un sous-répertoire de `createdir` nommé `attributes` et ajoutez un fichier nommé `default.rb` qui contient les définitions d'attribut.
4. Modifiez `.kitchen.yml` pour supprimer CentOS de la liste des plateformes.
5. Exécutez `kitchen converge`, puis connectez-vous aux instances et vérifiez que les répertoires `/srv/www/shared` et `/srv/www/config` sont là.

> **Note**
>
> Avec AWS OpsWorks Stacks, définir des valeurs sous forme d'attributs offre un avantage supplémentaire ; vous pouvez utiliser du JSON personnalisé pour remplacer ces valeurs par pile ou même par déploiement. Cela peut être utile pour différents objectifs, notamment les suivants :
>
> - Vous pouvez personnaliser le comportement de vos recettes, par exemple les paramètres de configuration ou les noms d'utilisateur, sans devoir modifier le livre de recettes.
>
> Vous pouvez, par exemple, utiliser le même livre de recettes pour différentes piles et le JSON personnalisé pour spécifier les paramètres de configuration clés pour une pile particulière. Cela vous permet d'économiser du temps et des efforts nécessaires pour modifier le livre de recettes ou d'utiliser un autre livre de recettes pour chaque pile.
> - Vous n'avez pas à mettre les informations potentiellement sensibles telles que les mots de passe de base de données dans votre référentiel de livres de recettes.
>
> Vous pouvez en revanche utiliser un attribut pour définir une valeur par défaut, puis utiliser le JSON personnalisé pour remplacer cette valeur par la valeur réelle.
>
> Pour plus d'informations sur l'utilisation du JSON personnalisé pour remplacer les attributs, consultez Remplacement des attributs.

Le fichier d'attribut est nommé `default.rb` parce que c'est une application Ruby, malgré sa simplicité. Cela signifie que vous pouvez, par exemple, utiliser la logique conditionnelle pour spécifier des valeurs d'attributs basées sur le système d'exploitation. Dans Logique conditionnelle, vous avez spécifié un nom de sous-répertoire différents pour différentes familles de Linux dans la recette. Avec un fichier d'attribut, vous pouvez plutôt mettre la logique conditionnelle dans le fichier d'attribut.

Le fichier d'attribut suivant utilise `value_for_platform` pour spécifier une autre valeur d'attribut `['shared_dir']` en fonction du système d'exploitation. Pour d'autres conditions, vous pouvez utiliser la logique `if-elsif-else` ou une instruction `case` de Ruby.

```
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'
```

Pour exécuter la recette

1. Exécutez `kitchen destroy` pour commencer avec une instance propre.
2. Remplacez le code de `attributes/default.rb` par l'exemple précédent.
3. Modifiez `.kitchen.yml` pour ajouter une plateforme CentOS à la section des plateformes, comme décrit dans Logique conditionnelle.
4. Exécutez `kitchen converge`, puis connectez-vous aux instances afin de vérifier que les répertoires sont là.

Une fois que vous avez terminé, exécutez `kitchen destroy` pour résilier l'instance. L'exemple suivant utilise un nouveau livre de recettes.

Exemple 6 : Création de fichiers

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une fois que vous avez créé des répertoires, vous devez souvent le remplir avec les fichiers de configuration, les fichiers de données, etc. Cette rubrique présente deux façons d'installer les fichiers sur une instance.

Rubriques

- Installation d'un fichier à partir d'un livre de recettes
- Création d'un fichier à partir d'un modèle

Installation d'un fichier à partir d'un livre de recettes

La façon la plus simple d'installer un fichier sur une instance est d'utiliser une ressource `cookbook_file` qui copie un fichier du livre de recettes dans un emplacement spécifié sur l'instance pour les systèmes Linux et Windows. Cet exemple étend la recette de Exemple 3 : Création de répertoires pour ajouter un fichier de données à `/srv/www/shared` après la création du répertoire. A titre de référence, voici la recette d'origine.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end
```

Pour configurer le livre de recettes

1. Dans le répertoire `opsworks_cookbooks`, créez un répertoire nommé `createfile` et accédez à celui-ci.
2. Ajoutez un fichier `metadata.rb` à `createfile` avec le contenu suivant.

   ```
   name "createfile"
   version "0.1.0"
   ```
3. Initialisez et configurez Test Kitchen, comme décrit dans Exemple 1 : Installation des packageset supprimez CentOS de la liste `platforms`.
4. Ajoutez un sous-répertoire `recipes` à `createfile`.

Le fichier à installer contient les données JSON suivantes.

```
{
  "my_name" : "myname",
  "your_name" : "yourname",
  "a_number" : 42,
  "a_boolean" : true
```

```
}
```

Pour configurer le fichier de données

1. Ajoutez un sous-répertoire `files` à `createfile` et un sous-répertoire `default` à `files`. N'importe quel fichier installé avec `cookbook_file` doit être dans un sous-répertoire de `files`, par exemple `files/default` dans cet exemple.

   > **Note**
   >
   > Si vous souhaitez spécifier des fichiers différents pour différents systèmes, vous pouvez mettre chaque fichier propre au système dans un sous-dossier nommé pour le système, par exemple `files/ubuntu`. La ressource `cookbook_file` copie le fichier propre au système approprié, s'il existe, et utilise le fichier `default` dans le cas contraire. Pour plus d'informations, consultez cookbook_file.

2. Créez un fichier nommé `example_data.json` avec le JSON de l'exemple précédent et ajoutez-le dans `files/default`.

La recette suivante copie `example_data.json` dans un emplacement spécifié.

```
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
```

Une fois que la ressource du répertoire a créé `/srv/www/shared`, la ressource `cookbook_file` copie `example_data.json` dans ce répertoire et définit également l'utilisateur, le groupe et le mode du fichier.

> **Note**
>
> La ressource `cookbook_file` présente une nouvelle action : `create_if_missing`. Vous pouvez également utiliser une action `create`, mais elle remplace un fichier existant. Si vous ne souhaitez pas remplacer quoi que ce soit, utilisez `create_if_missing`, qui installe `example_data.json` uniquement s'il n'existe pas déjà.

Pour exécuter la recette

1. Exécutez `kitchen destroy` pour commencer avec une instance propre.
2. Créez un fichier `default.rb` qui contient la recette précédente et enregistrez-le dans `recipes`.
3. Exécutez `kitchen converge`, puis connectez-vous à l'instance pour vérifier que `/srv/www/shared` contient le fichier `example_data.json`.

Création d'un fichier à partir d'un modèle

La ressource `cookbook_file` est utile dans certains cas, mais elle installe simplement les fichiers qui se trouvent dans le livre de recettes. Une ressource `template` offre un moyen plus souple d'installer un fichier sur une instance Windows ou Linux en le créant dynamiquement à partir d'un modèle. Vous pouvez ensuite déterminer les détails du contenu du fichier lors de l'exécution et les modifier en fonction des besoins. Par exemple, vous pouvez faire en sorte qu'un fichier de configuration ait un paramètre spécifique lorsque vous démarrez l'instance, puis modifier ce paramètre plus tard lorsque vous ajoutez des instances à la pile.

Cet exemple modifie le livre de recettes `createfile` de façon à utiliser une ressource `template` pour installer une version légèrement modifiée de `example_data.json`.

Voici à quoi ressemblera le fichier installé.

```
{
  "my_name" : "myname",
  "your_name" : "yourname",
  "a_number" : 42,
  "a_boolean" : true,
  "a_string" : "some string",
  "platform" : "ubuntu"
```

```
}
```

Les ressources de modèle sont généralement utilisés conjointement aux fichiers d'attributs, c'est pourquoi l'exemple en utilise un pour définir les valeurs suivantes.

```
default['createfile']['my_name'] = 'myname'
default['createfile']['your_name'] = 'yourname'
default['createfile']['install_file'] = true
```

Pour configurer le livre de recettes

1. Supprimez le répertoire `createfile` du livre de recettes `files` et son contenu.
2. Ajoutez un sous-répertoire `attributes` à `createfile` et ajoutez un fichier `default.rb` à `attributes` avec les définitions d'attributs précédentes.

Un modèle est un fichier `.erb` qui est globalement une copie du fichier final, avec une partie du contenu représenté par les espaces réservés. Lorsque la ressource `template` crée le fichier, elle copie le contenu du modèle dans le fichier spécifié et remplace les espaces réservés par leurs valeurs attribuées. Voici le modèle pour `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'] %>"
}
```

Les valeurs `<%=...%>` sont des espaces réservés.

- `<%=node[...]%>` représente une valeur d'attribut de nœud.

  Pour cet exemple, la valeur « your_name » est un espace réservé qui représente une des valeurs d'attributs du fichier d'attribut du livre de recettes.
- `<%=@...%>` représente la valeur d'une variable qui est définie dans le modèle de ressource, comme nous le verrons bientôt.

Pour créer le modèle de fichier

1. Ajoutez un sous-répertoire `templates` au livre de recettes `createfile` et un sous-répertoire `default` à `templates`.

   > **Note**
   >
   > Le répertoire `templates` fonctionne globalement comme le répertoire `files`. Vous pouvez mettre des modèles propres au système dans un sous-répertoire comme `ubuntu` qui est nommé pour le système. La ressource `template` utilise le modèle propre au système approprié s'il existe et elle utilise le modèle `default` s'il n'existe pas.

2. Créez un fichier appelé `example_data.json.erb` et placez-le dans le répertoire `templates/default`. Le nom du modèle est arbitraire, mais vous le créez généralement en ajoutant `.erb` au nom du fichier, avec les extensions.

La recette suivante utilise une ressource `template` pour créer `/srv/www/shared/example_data.json`.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end

template "/srv/www/shared/example_data.json" do
  source "example_data.json.erb"
  mode 0644
  variables(
    :a_boolean_var => true,
    :a_string_var => "some string"
  )
  only_if {node['createfile']['install_file']}
end
```

La ressource `template` crée `example_data.json` à partir d'un modèle et l'installe dans `/srv/www/shared`.

- Le nom du modèle, `/srv/www/shared/example_data.json`, spécifie le chemin et le nom du fichier installé.
- L'attribut `source` spécifie le modèle utilisé pour créer le fichier.
- L'attribut `mode` spécifie le mode du fichier installé.
- La ressource définit deux variables, `a_boolean_var` et `a_string_var`.

  Lorsque la ressource crée `example_data.json`, elle remplace les espaces réservés des variables du modèle par les valeurs correspondantes issues de la ressource.
- L'attribut `only_if` de protection demande à la ressource de créer le fichier uniquement si `['createfile']['install_file']` est défini sur `true`.

Pour exécuter la recette

1. Exécutez `kitchen destroy` pour commencer avec une instance propre.
2. Remplacez le code de `recipes/default.rb` par l'exemple précédent.
3. Exécutez `kitchen converge`, puis connectez-vous à l'instance pour vérifier que le fichier se trouve dans `/srv/www/shared` et a le contenu correct.

Lorsque vous avez terminé, exécutez `kitchen destroy` pour arrêter l'instance. La section suivante utilise un nouveau livre de recettes.

Exemple 7 : Exécution des commandes et des scripts

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les ressources de Chef peuvent gérer une grande variété de tâches sur une instance, mais il est parfois préférable d'utiliser une commande shell ou un script. Par exemple, vous utilisez peut-être certains scripts pour réaliser des tâches spécifiques et il sera plus facile de continuer à les utiliser

plutôt que d'implémenter un nouveau code. Cette section montre comment exécuter des commandes ou des scripts sur une instance.

Rubriques

- Exécution des commandes
- Exécution des scripts

Exécution des commandes

La ressource `script` exécute une ou plusieurs commandes. Elle prend en charge les interpréteurs de commande csh, bash, Perl, Python et Ruby afin de permettre une utilisation sur les systèmes Linux ou Windows tant que les interprètes appropriés sont installés. Cette rubrique montre comment exécuter une commande simple bash sur une instance Linux. Chef prend également en charge les ressources powershell_script et batch pour exécuter des scripts sous Windows. Pour de plus amples informations, veuillez consulter Exécution d'un PowerShell script Windows.

Mise en route

1. Dans le répertoire `opsworks_cookbooks`, créez un répertoire nommé `script` et accédez à celui-ci.
2. Ajoutez un fichier `metadata.rb` à `script` avec le contenu suivant.

   ```
   name "script"
   version "0.1.0"
   ```

3. Initialisez et configurez Test Kitchen, comme décrit dans Exemple 1 : Installation des packageset supprimez CentOS de la liste `platforms`.
4. Dans `script`, créez un répertoire nommé `recipes`.

Vous pouvez exécuter des commandes en utilisant la ressource `script` elle-même, mais Chef prend également en charge un ensemble de versions propres à l'interpréteur de commande de la ressource, qui sont nommées pour l'interpréteur. La recette suivante utilise une ressource `bash` pour exécuter un script bash simple.

```
bash "install_something" do
  user "root"
```

```
  cwd "/tmp"
  code <<-EOH
    touch somefile
  EOH
  not_if do
    File.exists?("/tmp/somefile")
  end
end
```

La ressource `bash` est configurée comme suit.

- Elle utilise l'action par défaut, `run`, qui exécute les commandes dans le bloc `code`.

  Cet exemple a une commande, `touch somefile`, mais un bloc `code` peut contenir plusieurs commandes.
- L'attribut `user` spécifie l'utilisateur qui exécute la commande.
- L'attribut `cwd` spécifie le répertoire de travail.

  Pour cet exemple, `touch` crée un fichier dans le répertoire `/tmp`.
- L'attribut de protection `not_if` indique à la ressource de ne rien faire si le fichier existe déjà.

Pour exécuter la recette

1. Créez un fichier `default.rb` qui contient l'exemple de code précédent et enregistrez-le dans `recipes`.
2. Exécutez `kitchen converge`, puis connectez-vous à l'instance pour vérifier que le fichier se trouve dans `/tmp`.

Exécution des scripts

La ressource `script` est pratique, surtout si vous avez besoin d'exécuter une ou deux commandes seulement, mais il est souvent préférable de stocker le script dans un fichier et d'exécuter le fichier. La ressource `execute` exécute un fichier exécutable spécifié, y compris les fichiers de script, sous Linux ou Windows. Cette rubrique modifie le livre de recettes `script` de l'exemple précédent afin d'utiliser `execute` pour exécuter un script shell simple. Vous pouvez facilement étendre l'exemple pour des scripts plus complexes ou d'autres types de fichiers exécutables.

Pour configurer le fichier de script

1. Ajoutez un sous-répertoire `files` à `script` et un sous-répertoire `default` à `files`.
2. Créez un fichier nommé `touchfile` qui contient les éléments suivants et ajoutez-le à `files/default`. Une ligne courante d'interpréteur Bash est utilisée dans cet exemple, mais vous pouvez opter pour un interpréteur qui fonctionne dans votre environnement shell si nécessaire.

   ```
   #!/usr/bin/env bash
   touch somefile
   ```

   Le fichier de script peut contenir n'importe quel nombre de commandes. Pour plus de facilité, cet exemple de script n'a qu'une seule commande `touch`.

La recette suivante exécute le script.

```
cookbook_file "/tmp/touchfile" do
  source "touchfile"
  mode 0755
end

execute "touchfile" do
  user "root"
  cwd "/tmp"
  command "./touchfile"
end
```

La ressource `cookbook_file` copie le fichier de script dans `/tmp` et définit le mode de façon à rendre le fichier exécutable. La ressource `execute` exécute ensuite le fichier comme suit :

- L'attribut `user` spécifie l'utilisateur de la commande (`root` dans cet exemple).
- L'attribut `cwd` spécifie le répertoire de travail (`/tmp` dans cet exemple).
- L'attribut `command` spécifie le script à exécuter (`touchfile` dans cet exemple) qui se trouve dans le répertoire de travail.

Pour exécuter la recette

1. Remplacez le code de `recipes/default.rb` par l'exemple précédent.

2. Exécutez `kitchen converge`, puis connectez-vous à l'instance pour vérifier que `/tmp` contient maintenant le fichier de script, avec le mode défini sur 0755 et `somefile`.

Lorsque vous avez terminé, exécutez `kitchen destroy` pour arrêter l'instance. La section suivante utilise un nouveau livre de recettes.

Exemple 8 : Gestion des services

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les packages tels que les serveurs d'application ont généralement un service associé qui doit être démarré, arrêtée, redémarré, etc. Par exemple, vous devez commencer le service Tomcat après avoir installé le package ou une fois que l'instance a terminé le démarrage, puis redémarrer le service chaque fois que vous modifiez le fichier de configuration. Cette rubrique présente les bases de la gestion d'un service sur une instance Linux, à l'aide d'un serveur d'application Tomcat à titre d'exemple. La ressource de service fonctionne de la même manière sur les instances Windows, bien qu'il existe quelques différences dans le détail. Pour de plus amples informations, veuillez consulter `service`.

> **Note**
>
> L'exemple est une installation de Tomcat très minimale, juste suffisante pour montrer les bases de l'utilisation d'une ressource `service`. Pour obtenir un exemple de l'implémentation des recettes afin d'obtenir un serveur Tomcat plus fonctionnel, consultez Création d'une couche serveur Tomcat personnalisée.

Rubriques

- Définition et démarrage d'un service
- Utilisation de notifies pour démarrer ou redémarrer un service

Définition et démarrage d'un service

Cette section montre les bases de la définition et du démarrage d'un service.

Mise en route

1. Dans le répertoire `opsworks_cookbooks`, créez un répertoire nommé `tomcat` et accédez à celui-ci.
2. Ajoutez un fichier `metadata.rb` à `tomcat` avec le contenu suivant.

   ```
   name "tomcat"
   version "0.1.0"
   ```

3. Initialisez et configurez Test Kitchen, comme décrit dans Exemple 1 : Installation des packageset supprimez CentOS de la liste `platforms`.
4. Ajoutez un sous-répertoire `recipes` à `tomcat`.

Vous utilisez une ressource `service` pour gérer un service. La recette par défaut suivante installe Tomcat et démarre le service.

```
execute "install_updates" do
  command "apt-get update"
end

package "tomcat7" do
    action :install
end

include_recipe 'tomcat::service'

service 'tomcat' do
  action :start
end
```

La recette exécute les tâches suivantes :

- La ressource `execute` exécute `apt-get update` pour installer les mises à jour actuelles du système.

Pour l'instance Ubuntu utilisée dans cet exemple, vous devez installer les mises à jour avant d'installer Tomcat. D'autres systèmes peuvent avoir des exigences différentes.

- La ressource `package` installe Tomcat 7.
- La recette `tomcat::service` incluse définit le service et nous reviendrons sur ce point plus tard.
- La ressource `service` démarre le service Tomcat.

  Vous pouvez également utiliser cette ressource afin d'émettre d'autres commandes, telles que l'arrêt et le redémarrage du service.

L'exemple suivant montre la recette `tomcat::service`.

```
service 'tomcat' do
  service_name "tomcat7"
  supports :restart => true, :reload => false, :status => true
  action :nothing
end
```

Cette recette crée la définition du service Tomcat comme suit :

- Le nom de la ressource, `tomcat`, est utilisé par d'autres recettes pour référencer le service.

  Par exemple, `default.rb` référence `tomcat` pour démarrer le service.
- La ressource `service_name` spécifie le nom du service.

  Lorsque vous listez les services sur l'instance, le service Tomcat est nommé tomcat7.
- `supports` spécifie la façon dont Chef gère les commandes `restart`, `reload` et `status` du service.
  - `true` indique que Chef peut utiliser le script init ou un autre fournisseur de services pour exécuter la commande.
  - `false` indique que le Chef doit tenter d'exécuter la commande par d'autres moyens.

Notez qu'`action` est définie sur `:nothing`, une valeur qui indique à la ressource de ne rien faire. La ressource de service prend en charge les actions telles que `start` et `restart`. Toutefois, ce livre de recettes suit une pratique standard d'utilisation d'une définition de service qui ne réalise aucune action et de démarrage ou de redémarrage du service ailleurs. Chaque recette qui démarre

ou redémarre un service doit d'abord le définir, ce qui signifie que l'approche la plus simple consiste à mettre la définition de service dans une recette distincte et à l'inclure dans les autres recettes en fonction des besoins.

> **Note**
>
> A des fins de simplicité, la recette par défaut pour cet exemple utilise une ressource `service` pour démarrer le service après l'exécution de la définition de service. Une implémentation de production démarre ou redémarre généralement un service à l'aide de `notifies`, comme indiqué plus tard.

Pour exécuter la recette

1. Créez un fichier `default.rb` qui contient l'exemple de recette par défaut et enregistrez-le dans `recipes`.
2. Créez un fichier `service.rb` qui contient l'exemple de définition de service et enregistrez-le dans `recipes`.
3. Exécutez `kitchen converge`, puis connectez-vous à l'instance et exécutez la commande suivante pour vérifier si le service est en cours d'exécution.

   ```
   sudo service tomcat7 status
   ```

> **Note**
>
> Si vous exécutiez `service.rb` séparément de `default.rb`, vous devrez modifier `.kitchen.yml` de façon à ajouter `tomcat::service` à la liste d'exécution. Toutefois, lorsque vous incluez une recette, son code est intégré dans la recette parent avant l'exécution de la recette. `service.rb` fait donc globalement partie de `default.rb` et n'a pas besoin d'une entrée de liste d'exécution distincte.

Utilisation de notifies pour démarrer ou redémarrer un service

En règle générale, l'implémentation de production n'utilise pas `service` pour démarrer ou redémarrer un service. En revanche, elle ajoute `notifies` à une ressource. Par exemple, si vous souhaitez redémarrer le service après avoir modifié un fichier de configuration, vous incluez

`notifies` dans la ressource `template` associée. L'utilisation de `notifies` a les avantages suivants par rapport à l'utilisation d'une ressource `service` pour redémarrer explicitement le service.

- L'élément `notifies` redémarre le service si le fichier de configuration associé a changé, il n'y a donc aucun risque de provoquer un redémarrage inutile du service.
- Chef redémarre le service au plus une fois à la fin de chaque exécution, quel que soit le nombre de `notifies` dans l'exécution.

  Par exemple, l'exécution de Chef peut inclure plusieurs ressources de modèle, chacune modifiant un fichier de configuration différent et nécessitant un redémarrage du service si le fichier a été modifié. Cependant, vous souhaitez généralement redémarrer le service une seule fois, à la fin de l'exécution de Chef. Sinon, vous pourriez tenter de redémarrer un service qui n'est pas encore entièrement opérationnel et fait partie d'un redémarrage antérieur, ce qui peut entraîner des erreurs.

Cet exemple modifie `tomcat::default` de façon à inclure une ressource `template` qui utilise `notifies` pour redémarrer le service. Un exemple réaliste utiliserait une ressource de modèle qui crée une version personnalisée de l'un des fichiers de configuration Tomcat, mais celles-ci sont assez longues et complexes. A des fins de simplicité, l'exemple utilise uniquement le modèle de ressource de Création d'un fichier à partir d'un modèle. Il n'a rien à voir avec Tomcat, mais il offre un moyen simple de montrer comment utiliser `notifies`. Pour obtenir un exemple de l'utilisation des modèles pour créer des fichiers de configuration Tomcat, consultez Recettes Setup.

Pour configurer le livre de recettes

1. Ajoutez un sous-répertoire `templates` à `tomcat` et un sous-répertoire `default` à `templates`.
2. Copiez le modèle `example_data.json.erb` du livre de recettes `createfile` dans le répertoire `templates/default`.
3. Ajoutez un sous-répertoire `attributes` à `tomcat`.
4. Copiez le fichier d'attribut `default.rb` du livre de recettes `createfile` dans le répertoire `attributes`.

La recette suivante utilise `notifies` pour redémarrer le service Tomcat.

```
execute "install_updates" do
  command "apt-get update"
```

```
end

package "tomcat7" do
    action :install
end

include_recipe 'tomcat::service'

service 'tomcat' do
  action :enable
end

directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end

template "/srv/www/shared/example_data.json" do
  source "example_data.json.erb"
  mode 0644
  variables(
    :a_boolean_var => true,
    :a_string_var => "some string"
  )
  only_if {node['createfile']['install_file']}
  notifies :restart, resources(:service => 'tomcat')
end
```

L'exemple fusionne la recette de Création d'un fichier à partir d'un modèle dans la recette de la section précédente, avec deux modifications importantes :

- La ressource `service` est toujours là, mais elle a désormais une utilité légèrement différente.

  L'action `:enable` active le service Tomcat au moment du démarrage.
- Le modèle de ressource inclut maintenant `notifies`, qui redémarre le service Tomcat si `example_data.json` a changé.

  De cette façon, le service est démarré lorsque Tomcat est installé pour la première fois et redémarré après chaque changement de configuration.

Pour exécuter la recette

1. Exécutez `kitchen destroy` pour commencer avec une instance propre.
2. Remplacez le code de `default.rb` par l'exemple précédent.
3. Exécutez `kitchen converge`, puis connectez-vous à l'instance et vérifiez que le service est en cours d'exécution.

> **Note**
>
> Si vous voulez redémarrer un service, mais que la recette n'inclut pas de ressource telle que `template` qui prend en charge `notifies`, vous pouvez utiliser une ressource `execute` factice à la place. Par exemple
>
> ```
> execute 'trigger tomcat service restart' do
>   command 'bin/true'
>   notifies :restart, resources(:service => 'tomcat')
> end
> ```
>
> La ressource `execute` doit avoir un attribut `command`, même si vous utilisez la ressource uniquement pour exécuter `notifies`. Cet exemple contourne cette exigence en exécutant `/bin/true`, une commande shell qui retourne simplement un code de réussite.

Exemple 9 : utilisation des EC2 instances Amazon

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Jusqu'à présent, vous avez exécuté des instances localement dans VirtualBox. Bien que cela soit simple et rapide, vous souhaiterez éventuellement tester vos recettes sur une EC2 instance

Amazon. En particulier, si vous souhaitez exécuter des recettes sur Amazon Linux, celles-ci ne sont disponibles que sur Amazon EC2. Vous pouvez utiliser un système similaire tel que CentOS pour la mise en œuvre et les tests préliminaires, mais le seul moyen de tester complètement vos recettes sur Amazon Linux est d'utiliser une instance Amazon EC2 .

Cette rubrique explique comment exécuter des recettes sur une EC2 instance Amazon. Vous allez utiliser Test Kitchen et Vagrant plus ou moins de la même manière que dans les sections précédentes, avec deux différences :

- Le pilote est `kitchen-ec2` au lieu de Vagrant.
- Le `.kitchen.yml` fichier du livre de recettes doit être configuré avec les informations requises pour lancer l' EC2 instance Amazon.

> **Note**
>
> Une autre approche consiste à utiliser le plug-in `vagrant-aws` de Vagrant. Pour plus d'informations, consultez Fournisseur AWS Vagrant.

Vous aurez besoin d'informations d'identification AWS pour créer une EC2 instance Amazon. Si vous n'avez pas de compte AWS, vous pouvez en obtenir un, comme suit.

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

   Une partie de la procédure d'inscription consiste à recevoir un appel téléphonique ou un message texte et à saisir un code de vérification sur le clavier du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à https://aws.amazon.com/et en choisissant Mon compte.

Création d'un utilisateur doté d'un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez Utilisateur racine d'un compte AWS AWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez Connexion en tant qu'utilisateur racine dans le Guide de l'utilisateur Connexion à AWS .
2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, consultez la section Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d' AWS IAM Identity Center dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu'utilisateur doté d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribution d'un accès à d'autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d'autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

   Pour obtenir des instructions, consultez Création d'un ensemble d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

   Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l'utilisateur AWS IAM Identity Center .

Vous devez créer un utilisateur IAM autorisé à accéder à Amazon EC2 et enregistrer les clés d'accès et secrètes de l'utilisateur dans un emplacement sécurisé sur votre poste de travail. Test Kitchen utilisera ces informations d'identification pour créer l'instance. Le meilleur moyen de fournir des informations d'identification pour Test Kitchen est d'affecter les clés aux variables d'environnement suivantes sur votre poste de travail.

> **Warning**
>
> Les utilisateurs IAM disposent d'informations d'identification à long terme, ce qui présente un risque de sécurité. Pour atténuer ce risque, nous vous recommandons de ne fournir à ces utilisateurs que les autorisations dont ils ont besoin pour effectuer la tâche et de supprimer ces autorisations lorsqu'elles ne sont plus nécessaires.

- AWS_ACCESS_KEY — la clé d'accès de votre utilisateur, qui AKIAIOSFODNN7EXAMPLE ressemblera à
- AWS_SECRET_KEY — la clé secrète de votre utilisateur, qui wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY ressemblera à

Cette approche réduit les risques de compromission accidentelle de votre compte, par exemple, en chargeant un projet qui contient vos informations d'identification dans un référentiel public.

Pour configurer le livre de recettes

1. Pour utiliser le pilote `kitchen-ec2`, le package `ruby-dev` doit être installé sur votre système. L'exemple de commande suivant montre comment utiliser `aptitude` pour installer le package sur un système Ubuntu.

   ```
   sudo aptitude install ruby1.9.1-dev
   ```

2. Le pilote `kitchen-ec2` est un GEM que vous pouvez installer comme suit :

   ```
   gem install kitchen-ec2
   ```

   En fonction de votre poste de travail, cette commande peut avoir besoin de `sudo`, ou vous pouvez également utiliser un gestionnaire de l'environnement Ruby comme RVM. Cette procédure a été testée avec la version 0.8.0 du pilote `kitchen-ec2` pilote, mais il y a des versions plus récentes. Pour installer une version spécifique, exécutez `gem install kitchen-ec2 -v <version number>`.

3. Vous devez spécifier une paire de clés Amazon EC2 SSH que Test Kitchen peut utiliser pour se connecter à l'instance. Si vous ne possédez pas de paire de EC2 clés Amazon, consultez Amazon EC2 Key Pairs pour savoir comment en créer une. Notez que la paire de clés doit appartenir à la même région AWS que l'instance. L'exemple utilise l'ouest des États-Unis (Californie du Nord).

   Une fois que vous avez sélectionné une paire de clés, créez un sous-répertoire d'`opsworks_cookbooks` nommé `ec2_keys` et copiez la clé privée de la paire de clés (`.pem`) dans ce sous-répertoire. Notez que le fait de placer la clé privée dans `ec2_keys` permet uniquement de simplifier légèrement le code, mais que cette clé peut se trouver n'importe où sur votre système.

4. Créez un sous-répertoire d'`opsworks_cookbooks`, nommé `createdir-ec2` et accédez à celui-ci.

5. Ajoutez un fichier `metadata.rb` à `createdir-ec2` avec le contenu suivant.

   ```
   name "createdir-ec2"
   ```

```
version "0.1.0"
```

6. Initialisez Test Kitchen, comme décrit dans Exemple 1 : Installation des packages. La section suivante décrit la procédure de configuration `.kitchen.yml`, ce qui est nettement plus compliqué pour les EC2 instances Amazon.
7. Ajoutez un sous-répertoire `recipes` à `createdir-ec2`.

Configuration de .kitchen.yml pour Amazon EC2

Vous configurez `.kitchen.yml` avec les informations dont le `kitchen-ec2` pilote a besoin pour lancer une EC2 instance Amazon correctement configurée. Voici un exemple de `.kitchen.yml` fichier pour une instance Amazon Linux dans la région de l'ouest des États-Unis (Californie du Nord).

```
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:
```

Vous pouvez utiliser les paramètres par défaut pour les sections `provisioner` et `suites`, mais vous devez modifier les paramètres par défaut `driver` et `platforms`. Cet exemple utilise une liste minimale de paramètres et accepte les valeurs par défaut pour le reste. Pour une liste complète des `kitchen-ec2` paramètres, consultez Kitchen : :Ec2 : A Test Kitchen Driver for Amazon. EC2

L'exemple définit les attributs `driver` suivants. Il part du principe que vous avez attribué des clés secrètes et d'accès de l'utilisateur aux variables d'environnement standard, comme indiqué plus tôt. Le pilote utilise ces clés par défaut. Sinon, vous devez spécifier explicitement les clés en ajoutant `aws_access_key_id` et `aws_secret_access_key` aux attributs `driver`, définis sur les valeurs clés appropriées.

name

(Obligatoire) Cet attribut doit être défini sur `ec2`.

aws_ssh_key_id

(Obligatoire) Le nom de la paire de clés Amazon EC2 SSH, nommé `US-East1` dans cet exemple.

transport.ssh_key

(Obligatoire) Le fichier de clé privée (`.pem`) pour la clé que vous avez spécifiée pour `aws_ssh_key_id`. Pour cet exemple, le fichier est nommé `US-East1.pem` et se trouve dans le répertoire `../opsworks/ec2_keys`.

region

(Obligatoire) La région AWS de l'instance. L'exemple utilise l'ouest des États-Unis (Californie du Nord), représenté par`us-west-1`).

availability_zone

(Facultatif) La zone de disponibilité de l'instance. Si vous omettez ce paramètre, Test Kitchen utilise une zone de disponibilité par défaut pour la région spécifiée, à savoir l'ouest `us-west-1b` des États-Unis (Californie du Nord). Toutefois, il est possible que la zone par défaut ne soit pas disponible pour votre compte. Dans ce cas, vous devez spécifier explicitement une zone de disponibilité. Il se trouve que le compte utilisé pour préparer les exemples ne prend pas en charge `us-west-1b` et l'exemple spécifie explicitement `us-west-1c`.

require_chef_omnibus

Lorsqu'il est défini sur `true`, ce paramètre veille à ce que le programme d'installation général soit utilisé pour installer `chef-client` sur toutes les instances de la plateforme.

security_group_ids

(Facultatif) Liste des groupes de sécurité IDs à appliquer à l'instance. Ce paramètre applique le groupe de sécurité `default` à l'instance. Assurez-vous que les règles de trafic entrant des groupes de sécurité autorisent les connexions SSH entrantes, ou que Test Kitchen ne soit pas en mesure de communiquer avec l'instance. Si vous utilisez le groupe de sécurité `default`, vous devrez peut-être le modifier en conséquence. Pour plus d'informations, consultez Amazon EC2 Security Groups.

subnet_id

L'ID du sous-réseau cible de l'instance, le cas échéant.

associate_public_ip

Vous pouvez demander EC2 à Amazon d'associer une adresse IP publique à l'instance si vous souhaitez pouvoir accéder à l'instance depuis Internet.

interface

Le type de configuration du nom d'hôte que vous utilisez pour accéder à l'instance. Les valeurs valides sont `dns`, `public`, `private` ou `private_dns`. Si vous ne spécifiez pas de valeur pour cet attribut, `kitchen-ec2` définit la configuration du nom d'hôte dans l'ordre suivant. Si vous omettez cet attribut, le type de configuration n'est pas défini.

1. Nom du DNS
2. Adresse IP publique
3. Adresse IP privée
4. Nom DNS privé

Important

Plutôt que d'utiliser les informations d'identification de votre compte pour les clés d'accès et les clés secrètes, vous devez créer un utilisateur et fournir ces informations d'identification à Test Kitchen. Pour plus d'informations, consultez Bonnes pratiques en matière de gestion des clés d'accès AWS.

Veillez à ne pas le placer `.kitchen.yml` dans un emplacement accessible au public, par exemple en le téléchargeant dans un dépôt public GitHub ou Bitbucket. Vous risqueriez d'exposer vos informations d'identification et de compromettre la sécurité de votre compte.

Le pilote `kitchen-ec2` fournit la prise en charge par défaut pour les plateformes suivantes :

- 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

Si vous souhaitez utiliser une ou plusieurs de ces plateformes, ajoutez les noms de plateforme appropriés à `platforms`. Le pilote `kitchen-ec2` sélectionne automatiquement une AMI appropriée et génère un nom d'utilisateur SSH. Vous pouvez utiliser d'autres plateformes (cet exemple utilise Amazon Linux), mais vous devez spécifier explicitement les attributs suivants. `platforms`

name

Le nom de la plateforme. Cet exemple utilise Amazon Linux, donc `name` est défini sur `amazon`.

driver

Les attributs `driver`, qui incluent les éléments suivants :

- `image_id`— L'AMI de la plateforme, qui doit appartenir à la région spécifiée. L'exemple utilise `ami-ed8e9284` une AMI Amazon Linux de la région de l'ouest des États-Unis (Californie du Nord).
- `transport.username`— Le nom d'utilisateur SSH que Test Kitchen utilisera pour communiquer avec l'instance.

  Utilisez `ec2-user` pour Amazon Linux. D'autres AMIs peuvent avoir des noms d'utilisateur différents.

Remplacez le code de `.kitchen.yml` par l'exemple et attribuez les valeurs appropriées à des attributs propres au compte, par exemple `aws_access_key_id`.

Exécution de la recette

Cet exemple utilise la recette de Itération.

Pour exécuter la recette

1. Créez un fichier nommé `default.rb` avec le code suivant et enregistrez-le dans le dossier `recipes` du livre de recettes.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end
```

2. Exécutez `kitchen converge` pour exécuter la recette. Notez que l'exécution de cette commande prendra plus de temps que dans les exemples précédents en raison du temps nécessaire au lancement et à l'initialisation d'une EC2 instance Amazon.
3. Accédez à la EC2console Amazon, sélectionnez la région USA Ouest (Californie du Nord), puis cliquez sur Instances dans le volet de navigation. Vous verrez l'instance nouvellement créée dans la liste.
4. Exécutez `kitchen login` pour vous connecter à l'instance, comme vous l'avez fait pour les instances en cours d'exécution VirtualBox. Vous verrez les répertoires nouvellement créés sous `/srv`. Vous pouvez également utiliser votre client SSH pour vous connecter à l'instance.

Étapes suivantes

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Ce chapitre a déjà présenté les bases de l'implémentation des livres de recettes de Chef, mais ce n'est pas tout :

- Les exemples vous ont montré comment utiliser certaines des ressources les plus couramment utilisées, mais ce n'est pas tout.

  Pour les ressources qui ont été couvertes, les exemples ont utilisé uniquement certains des attributs et des actions disponibles. Pour une référence complète, consultez À propos des ressources et des fournisseurs.
- Les exemples ont utilisé uniquement les éléments des livre de recettes de base : `recipes`, `attributes` `files` et `templates`.

  Les livres de recettes peuvent aussi inclure divers autres éléments, tels que `libraries`, `definitions` et `specs`. Pour plus d'informations, consultez la documentation de Chef.
- Les exemples ont utilisé Test Kitchen afin de faciliter le démarrage des instances, l'exécution des recettes et la connexion aux instances.

  Test Kitchen est principalement une plateforme de test que vous pouvez utiliser pour exécuter divers tests sur vos recettes. Si vous ne l'avez pas déjà fait, parcourez le reste de la procédure Test Kitchen, qui présente ses fonctionnalités de test.
- Implémentation de Cookbooks for Stacks AWS OpsWorksfournit des exemples plus avancés et montre comment implémenter des livres de recettes pour AWS OpsWorks Stacks.

Implémentation de Cookbooks for Stacks AWS OpsWorks

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Principes de base des livre de recettes vous a présenté les livres de recettes et les recettes. Les exemples de cette section étaient simples par conception et fonctionneront sur toutes les instances compatibles avec Chef, y compris les instances AWS OpsWorks Stacks. Pour implémenter des

livres de recettes plus sophistiqués pour AWS OpsWorks Stacks, vous devez généralement tirer pleinement parti de l'environnement AWS OpsWorks Stacks, qui diffère de Chef standard à bien des égards.

Cette rubrique décrit les principes de base de l'implémentation de recettes pour les instances AWS OpsWorks Stacks.

> **Note**
>
> Si vous n'êtes pas encore à l'aise avec l'implémentation des livres de recettes, commencez par Principes de base des livre de recettes.

Rubriques

- Exécution d'une recette sur une AWS OpsWorks instance Stacks Linux
- Exécution d'une recette sur une instance Windows
- Exécution d'un PowerShell script Windows
- Simulation des attributs de configuration et de déploiement de la pile sur Vagrant
- Utilisation des valeurs des attributs de configuration et de déploiement de la pile
- Utilisation d'un livre de recettes externe sur une instance Linux : Berkshelf
- Utilisation du SDK pour Ruby : téléchargement de fichiers depuis Amazon S3
- Installation de logiciels Windows
- Remplacement des attributs intégrés
- Remplacement des modèles intégrés

Exécution d'une recette sur une AWS OpsWorks instance Stacks Linux

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Test Kitchen et Vagrant fournissent un moyen simple et efficace d'implémenter des livres de recettes, mais pour vérifier que les recettes d'un livre de recettes fonctionneront correctement en production, vous devez les exécuter sur une AWS OpsWorks instance Stacks. Cette rubrique décrit comment installer un livre de recettes personnalisé sur une instance Linux AWS OpsWorks Stacks et exécuter une recette simple. Elle propose également des conseils pour corriger efficacement les bogues de la recette.

Pour une description de l'exécution des recettes sur des instances Windows, consultez Exécution d'une recette sur une instance Windows.

Rubriques

- Création et exécution de la recette
- Exécution de la recette automatiquement
- Dépannage et réparation des recettes

Création et exécution de la recette

Tout d'abord, vous devez créer une pile. Voici un bref résumé de la création d'une pile pour cet exemple. Pour de plus amples informations, veuillez consulter Créer une pile.

Pour créer une pile

1. Ouvrez la console AWS OpsWorks Stacks et cliquez sur Add Stack (Ajouter une pile).
2. Spécifiez les paramètres suivants, acceptez les valeurs par défaut pour les autres paramètres et cliquez sur Add Stack (Ajouter une pile).

   - Nom — OpsTest
   - Clé SSH par défaut : une paire de EC2 clés Amazon

   Si vous devez créer une paire de EC2 clés Amazon, consultez Amazon EC2 Key Pairs. Notez que la paire de clés doit appartenir à la même région AWS que l'instance. L'exemple utilise la région USA Ouest (Oregon) par défaut.
3. Cliquez sur Add a layer (Ajouter une couche) et ajoutez une couche personnalisée à la pile avec les paramètres suivants.

   - Nom — OpsTest
   - Nom abrégé — opstest

N'importe quel type de couche peut être utilisé pour les piles Linux, mais l'exemple n'a pas besoin des packages installés par les autres types de couches, c'est pourquoi une couche personnalisée constitue l'approche la plus simple.

4. Ajoutez une instance 24/7 avec les paramètres par défaut de la couche et démarrez-la.

Pendant le démarrage de l'instance (cela prend généralement plusieurs minutes), vous pouvez créer le livre de recettes. Cet exemple utilise une version légèrement modifiée de la recette de Logique conditionnelle, qui crée un répertoire de données dont le nom dépend de la plateforme.

Pour configurer le livre de recettes

1. Créez un répertoire dans `opsworks_cookbooks`, nommé `opstest` et accédez à celui-ci.
2. Créez un fichier `metadata.rb` avec le contenu suivant et enregistrez-le sur `opstest`.

```
name "opstest"
version "0.1.0"
```

3. Créez un répertoire `recipes` dans `opstest`.
4. Créez un fichier `default.rb` avec la recette suivante et enregistrez-le dans le répertoire `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
```

Notez que la recette consigne un message, mais en appelant `Chef::Log.info`. Vous n'utilisez pas Test Kitchen pour cet exemple, la `log` méthode n'est donc pas très utile. `Chef::Log.info`place le message dans le journal Chef, que vous pouvez lire une fois l'exécution de Chef terminée. AWS OpsWorks Stacks fournit un moyen facile de consulter ces journaux, comme décrit plus loin.

> **Note**
>
> Les journaux de Chef contiennent généralement beaucoup d'informations courantes et peu intéressantes. Les caractères « * » qui entourent le texte du message permettent de le repérer facilement.

5. Créez une archive `.zip` de `opsworks_cookbooks`. Pour installer votre livre de recettes sur une instance AWS OpsWorks Stacks, vous devez le stocker dans un référentiel et fournir à AWS OpsWorks Stacks les informations nécessaires pour télécharger le livre de recettes sur l'instance. Vous pouvez stocker vos livres de recettes sur plusieurs types de référentiels pris en charge. Cet exemple stocke un fichier d'archive contenant les livres de recettes dans un compartiment Amazon S3. Pour plus d'informations sur les référentiels de livre de recettes, consultez Référentiels de livres de recettes.

   > **Note**
   >
   > Pour plus de simplicité, cet exemple archive simplement l'ensemble du répertoire `opsworks_cookbooks`. Cependant, cela signifie que AWS OpsWorks Stacks téléchargera tous les livres de recettes dans `opsworks_cookbooks` l'instance, même si vous n'utiliserez qu'un seul d'entre eux. Pour installer uniquement l'exemple de livre de recettes, créez un autre répertoire parent et transférez `opstest` dans ce répertoire. Créez ensuite une archive `.zip` du répertoire parent et utilisez-la au lieu d'`opsworks_cookbooks.zip`.
   >
   > Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

6. Téléchargez l'archive dans un compartiment Amazon S3, rendez-la publique et enregistrez l'URL de l'archive.

Vous pouvez maintenant installer le livre de recettes et exécuter la recette.

Pour exécuter la recette

1. Modifiez la pile pour activer les livres personnalisés et spécifiez les paramètres suivants.

   - Type de référentiel — S3 Archive
   - URL du référentiel : URL de l'archive du livre de recettes que vous avez enregistrée précédemment

   Utilisez les valeurs par défaut pour les autres paramètres, puis cliquez sur Save (Enregistrer) pour mettre à jour la configuration de la pile.
2. Exécutez la commande de pile de mise à jour des livres de recettes personnalisés, qui installe la version actuelle de vos livres de recettes personnalisés sur les instances de la pile. Si une version antérieure de vos livres de recettes est présente, cette commande la remplace.
3. Exécutez la recette à l'aide de la commande de pile Execute Recipes (Exécuter les recettes) après avoir défini Recipes to execute (Recettes à exécuter) sur **`opstest::default`**. Cette commande lance une exécution de Chef, avec une liste d'exécution composée de `opstest::default`.

Une fois que la recette a été exécutée correctement, vous pouvez la vérifier.

Pour vérifier opstest

1. La première étape consiste à examiner le journal de Chef. Cliquez sur show (afficher) dans la colonne Log (Journal) de l'instance opstest1. Faites défiler vers le bas pour afficher votre message de journal proche de la fin.

```
...
[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. Utilisez SSH pour vous connecter à l'instance et affichez la liste du contenu de `/srv/www/`.

Si vous avez suivi toutes les étapes, vous verrez `/srv/www/config` plutôt que le répertoire `/srv/www/shared` auquel vous vous attendiez. La section suivante propose des conseils pour corriger rapidement ce type de bogue.

Exécution de la recette automatiquement

La commande Execute Recipes (Exécuter les recettes) permet de tester aisément des recettes personnalisées. Elle est utilisée à cet effet dans la plupart de ces exemples. Toutefois, dans la pratique, vous exécutez généralement des recettes à des étapes standard du cycle de vie d'une instance, par exemple une fois le démarrage de l'instance terminé ou lorsque vous déployez une application. AWS OpsWorks Stacks simplifie l'exécution de recettes sur votre instance en prenant en charge un ensemble d'événements du cycle de vie pour chaque couche : installation, configuration, déploiement, dédéploiement et arrêt. Vous pouvez demander à AWS OpsWorks Stacks d'exécuter automatiquement une recette sur les instances d'une couche en affectant la recette à l'événement du cycle de vie approprié.

Vous devez généralement créer des répertoires dès qu'une instance a fini son démarrage, ce qui correspond à l'événement Setup. Voici comment exécuter l'exemple de recette lors de la configuration, à l'aide de la même pile que celle créée plus tôt dans l'exemple. Vous pouvez utiliser la même procédure pour les autres événements.

Pour exécuter automatiquement une recette lors de la configuration

1. Choisissez Layers dans le volet de navigation, puis cliquez sur l'icône en forme de crayon à côté du lien Recipes de la OpsTest couche.
2. Ajoutez **`opstest::default`** aux recettes Setup de la couche, cliquez sur + pour l'ajouter à la couche, puis choisissez Save (Enregistrer) pour enregistrer la configuration.
3. Choisissez Instances, ajoutez une autre instance à la couche et démarrez-la.

   L'instance doit être nommée `opstest2`. Une fois le démarrage terminé, AWS OpsWorks Stacks s'exécutera. `opstest::default`
4. Une fois l'instance `opstest2` en ligne, vérifiez que `/srv/www/shared` est présent.

> **Note**
>
> Si vous avez attribué des recettes aux événements Setup, Configure ou Deploy, vous les exécutez aussi manuellement en utilisant une commande de pile (Setup ou Configure) ou une commande de déploiement (Deploy) pour déclencher l'événement. Notez que si vous avez plusieurs recettes attribuées à un événement, ces commandes les exécutent toutes.

Dépannage et réparation des recettes

Si vous n'obtenez pas les résultats attendus ou si l'exécution de vos recettes échoue, commencez par examiner le journal de Chef pour chercher d'où vient le problème. Il contient une description détaillée de l'exécution et inclut des messages de journal en ligne de vos recettes. Les journaux sont particulièrement utiles si votre recette a simplement échoué. Lorsque cela se produit, Chef enregistre l'erreur, y compris une trace de la pile.

Si la recette a réussi, comme dans le cas de cet exemple, le journal de Chef ne vous sera pas très utile. Vous pouvez alors trouver d'où vient le problème en observant simplement la recette, et plus particulièrement les premières lignes :

```
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 est une alternative intéressante pour Amazon Linux lorsque vous testez des recettes sur Vagrant, mais vous exécutez actuellement les recettes sur une véritable instance Amazon Linux. La valeur de la plateforme pour Amazon Linux est `amazon`, qui n'est pas incluse dans l'appel `value_for_platform`, c'est pourquoi la recette crée `/srv/www/config` par défaut. Pour plus d'informations sur le dépannage, consultez Guide de débogage et dépannage.

Maintenant que vous avez identifié le problème, vous devez mettre à jour de la recette et vérifier le correctif. Vous pouvez revenir aux fichiers source d'origine, les mettre à jour`default.rb`, télécharger une nouvelle archive sur Amazon S3, etc. Toutefois, ce processus peut être légèrement

fastidieux et long. Voici une approche nettement plus rapide qui est particulièrement utile pour les bogues de recettes simples comme celui de l'exemple : modifier la recette sur l'instance.

Pour modifier une recette sur une instance

1. Utilisez SSH pour vous connecter à l'instance, puis exécutez `sudo su` pour élever vos privilèges. Vous avez besoin de privilèges racine pour accéder aux répertoires de livres de recettes.
2. AWS OpsWorks Stacks stocke votre livre de recettes`/opt/aws/opsworks/current/site-cookbooks`, alors naviguez vers. `/opt/aws/opsworks/current/site-cookbooks/opstest/recipes`

   > **Note**
   >
   > AWS OpsWorks Stacks stocke également une copie de vos livres de cuisine dans. `/opt/aws/opsworks/current/merged-cookbooks` Ne modifiez pas ce livre de recettes. Lorsque vous exécutez la recette, AWS OpsWorks Stacks copie le livre de recettes de `.../site-cookbooks` à`.../merged-cookbooks`, de sorte que toutes les modifications que vous apportez `.../merged-cookbooks` seront annulées.

3. Utilisez un éditeur de texte sur l'instance pour modifier `default.rb` et remplacez `centos` par `amazon`. Votre recette doit maintenant avoir l'aspect suivant.

```
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"
)
...
```

Pour vérifier le correctif, exécutez la recette à l'aide de la commande de pile Execute Recipe (Exécuter la recette). L'instance doit maintenant avoir un répertoire `/srv/www/shared`. Si vous avez besoin d'apporter des modifications supplémentaires à la recette, vous pouvez exécuter la commande Execute Recipe (Exécuter la recette) aussi souvent que vous le souhaitez. Vous n'avez pas besoin d'arrêter et redémarrer l'instance chaque fois que vous exécutez cette commande. Une

fois que la recette fonctionne correctement, n'oubliez pas de mettre à jour le code dans votre livre de recettes source.

> **Note**
>
> Si vous avez assigné votre recette à un événement du cycle de vie afin que AWS OpsWorks Stacks l'exécute automatiquement, vous pouvez toujours utiliser Execute Recipe pour réexécuter la recette. Vous pouvez également réexécuter la recette autant de fois que vous le souhaitez sans redémarrer l'instance en utilisant la console AWS OpsWorks Stacks pour déclencher manuellement l'événement approprié. Cependant, cette approche exécute toutes les recettes de l'événement. Voici un rappel :
>
> - Utilisez une commande de pile pour déclencher les événements Setup ou Configure.
> - Utilisez une commande de déploiement pour déclencher des événements Deploy ou Undeploy.

Exécution d'une recette sur une instance Windows

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette rubrique est en fait une version abrégée de Exécution d'une recette sur une instance Linux, qui vous montre comment exécuter une recette sur une pile Windows. Nous vous recommandons de commencer par parcourir Exécution d'une recette sur une instance Linux qui propose une description plus détaillée convenant aux deux types de systèmes d'exploitation.

Pour une description de la façon d'exécuter des recettes sur des instances AWS OpsWorks Stacks Linux, consultezExécution d'une recette sur une instance Linux.

Rubriques

- Activation de l'accès RDP

- Création et exécution de la recette
- Exécution de la recette automatiquement

Activation de l'accès RDP

Avant de commencer, si vous ne l'avez pas déjà fait, vous devez configurer un groupe de sécurité avec une règle entrante qui autorise l'accès RDP pour vos instances. Vous aurez besoin de ce groupe lorsque vous créerez la pile.

Lorsque vous créez la première pile dans une région, AWS OpsWorks Stacks crée un ensemble de groupes de sécurité. Ils en incluent un nommé quelque chose comme`AWS-OpsWorks-RDP-Server`, que AWS OpsWorks Stacks attache à toutes les instances Windows pour permettre l'accès RDP. Cependant, comme, par défaut, ce groupe de sécurité n'a pas de règles, vous devez ajouter une règle entrante pour autoriser l'accès RDP à vos instances.

Pour autoriser l'accès RDP

1. Ouvrez la EC2console Amazon, définissez-la sur la région de la pile et choisissez Security Groups dans le volet de navigation.
2. Choisissez AWS- OpsWorks -RDP-Server, choisissez l'onglet Inbound, puis sélectionnez Modifier.
3. Ajoutez une règle avec les paramètres suivants :
   - Type — RDP
   - Source — Les adresses IP sources autorisées.

     Généralement, vous autorisez les demandes RDP entrantes de votre adresse IP ou plages d'adresses IP spécifiée (habituellement, votre plage d'adresses IP d'entreprise).

> **Note**
>
> Comme indiqué plus tard, vous devez également modifier les autorisations utilisateur afin d'autoriser l'accès RDP pour les utilisateurs standard.

Pour de plus amples informations, veuillez consulter Connexion avec RDP.

Création et exécution de la recette

Voici un bref résumé de la création d'une pile pour cet exemple. Pour de plus amples informations, veuillez consulter Créer une pile.

Création d'une pile

1. Ouvrez la console AWS OpsWorks Stacks et choisissez Add Stack (Ajouter une pile). Spécifiez les paramètres suivants, acceptez les valeurs par défaut pour les autres paramètres et choisissez Add Stack (Ajouter une pile).

   - Nom — WindowsRecipeTest
   - Région — Ouest des États-Unis (Oregon)

     Cet exemple fonctionne dans n'importe quelle région, mais nous vous recommandons d'utiliser US West (Oregon) pour les didacticiels.
   - Système d'exploitation par défaut : Microsoft Windows Server 2012 R2
2. Choisissez Add a layer (Ajouter une couche) et ajoutez une couche personnalisée à la pile avec les paramètres suivants.

   - Nom — RecipeTest
   - Nom abrégé — recipetest
3. Ajoutez une instance 24 heures sur 24, 7 jours sur 7 avec les paramètres par défaut à la RecipeTest couche et démarrez-la.

   AWS OpsWorks Stacks attribue `AWS-OpsWorks-RDP-Server` automatiquement cette instance, ce qui permet aux utilisateurs autorisés de se connecter à l'instance.
4. Choisissez Permissions (Autorisations), puis Edit (Modifier), et choisissez SSH/RDP et sudo/admin. Les utilisateurs standard ont besoin de cette autorisation en plus du groupe de sécurité `AWS-OpsWorks-RDP-Server` pour se connecter à l'instance.

   > **Note**
   >
   > Vous pouvez également vous connecter en tant qu'Administrator, mais en ayant recours à une procédure différente. Pour de plus amples informations, veuillez consulter Connexion avec RDP.

Pendant le démarrage de l'instance (cela prend généralement plusieurs minutes), vous pouvez créer le livre de recettes. La recette de cet exemple crée un répertoire de données et est en fait la recette de Exemple 3 : Création de répertoires, modifiée pour Windows.

> **Note**
>
> Lorsque vous implémentez des livres de recettes pour des instances AWS OpsWorks Stacks Windows, vous utilisez une structure de répertoire quelque peu différente de celle que vous utilisez lorsque vous implémentez des livres de recettes pour des instances AWS OpsWorks Stacks Linux. Pour de plus amples informations, veuillez consulter Référentiels de livres de recettes.

Pour configurer le livre de recettes

1. Créez un répertoire nommé `windowstest` et accédez à celui-ci.
2. Créez un fichier `metadata.rb` avec le contenu suivant et enregistrez-le sur `windowstest`.

   ```
   name "windowstest"
   version "0.1.0"
   ```

3. Créez un répertoire `recipes` dans `windowstest`.
4. Créez un fichier `default.rb` avec la recette suivante et enregistrez-le dans le répertoire `recipes`.

   ```
   Chef::Log.info("******Creating a data directory.******")

   directory 'C:\data' do
     rights :full_control, 'instance_name\username'
     inherits false
     action :create
   end
   ```

   Remplacez *`username`* par votre nom d'utilisateur.

5. Placez le livre de recettes dans un référentiel.

Pour installer votre livre de recettes sur une instance AWS OpsWorks Stacks, vous devez le stocker dans un référentiel et fournir à AWS OpsWorks Stacks les informations nécessaires pour télécharger le livre de recettes sur l'instance. Vous pouvez stocker les livres de recettes Windows en tant que fichier d'archive dans un référentiel Git ou dans un compartiment S3. Cet exemple utilise un compartiment S3, c'est pourquoi vous devez créer une archive .zip du répertoire `windowstest`. Pour plus d'informations sur les référentiels de livre de recettes, consultez Référentiels de livres de recettes.

6. Chargez l'archive dans un compartiment S3, rendez l'archive publique et enregistrez l'URL de l'archive. Vous pouvez également utiliser une archive privée, mais une archive publique est suffisante pour cet exemple et légèrement plus facile à utiliser.

   Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Vous pouvez maintenant installer le livre de recettes et exécuter la recette.

Pour exécuter la recette

1. Modifiez la pile pour activer les livres personnalisés et spécifiez les paramètres suivants.

   - Type de référentiel — S3 Archive
   - URL du référentiel : URL de l'archive du livre de recettes que vous avez enregistrée précédemment

   Acceptez les valeurs par défaut pour les autres paramètres, puis choisissez Save (Enregistrer) pour mettre à jour la configuration de la pile.
2. Exécutez la commande de pile de mise à jour des livres de recettes personnalisés, qui installe la version actuelle de vos livres de recettes personnalisés sur les instances de la pile, dont les instances en ligne. Si une version antérieure de vos livres de recettes est présente, cette commande la remplace.
3. Une fois la mise à jour des livres de recettes personnalisées terminée, exécutez la recette en exécutant la commande de pile Execute Recipes (Exécuter les recettes) avec le paramètre Recipes to execute (Recettes à exécuter) défini sur **`windowstest::default`**. Cette commande lance une exécution de Chef, avec une liste d'exécution composée de votre recette.

Une fois que la recette a été exécutée correctement, vous pouvez la vérifier.

Pour vérifier windowstest

1. Examinez le journal de Chef. Choisissez show (afficher) dans la colonne Log (Journal) de l'instance opstest1. Faites défiler vers le bas pour afficher votre message de journal proche de la fin.

```
...
[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. Choisissez Instances, choisissez rdp dans la colonne Actions de l'instance et demandez un mot de passe RDP avec une date d'expiration adaptée. Copiez le nom DNS, le nom d'utilisateur et le mot de passe. Vous pouvez ensuite peuvent utiliser ces informations avec un client RDP, tel que le client de connexion Bureau à distance Windows, pour vous connecter à l'instance et vérifier que `c:\data` existe. Pour de plus amples informations, veuillez consulter Connexion avec RDP.

> **Note**
>
> Si votre recette ne fonctionne pas correctement, consultez Dépannage et réparation des recettes pour accéder aux astuces de dépannage, la plupart d'entre elles s'appliquant également aux instances Windows. Si vous souhaitez tester votre solution en modifiant la recette sur l'instance, recherchez votre livre de recettes dans le `C:\chef\cookbooks` répertoire, où AWS OpsWorks Stacks installe les livres de recettes personnalisés.

Exécution de la recette automatiquement

La commande Execute Recipes (Exécuter les recettes) permet de tester aisément des recettes personnalisées. Elle est utilisée à cet effet dans la plupart de ces exemples. Toutefois, dans la pratique, vous exécutez généralement des recettes à des étapes standard du cycle de vie d'une

instance, par exemple une fois le démarrage de l'instance terminé ou lorsque vous déployez une application. AWS OpsWorks Stacks simplifie l'exécution de recettes sur votre instance en prenant en charge un ensemble d'événements du cycle de vie pour chaque couche : installation, configuration, déploiement, dédéploiement et arrêt. Vous pouvez demander à AWS OpsWorks Stacks d'exécuter automatiquement une recette sur les instances d'une couche en affectant la recette à l'événement du cycle de vie approprié.

Vous devez généralement créer des répertoires dès qu'une instance a fini son démarrage, ce qui correspond à l'événement Setup. Voici comment exécuter l'exemple de recette lors de la configuration, à l'aide de la même pile que celle créée plus tôt dans l'exemple. Vous pouvez utiliser la même procédure pour les autres événements.

Pour exécuter automatiquement une recette lors de la configuration

1. Choisissez Layers dans le volet de navigation, puis cliquez sur l'icône en forme de crayon à côté du lien Recipes de la RecipeTest couche.
2. Ajoutez **`windowstest::default`** aux recettes Setup de la couche, choisissez + pour l'ajouter à la couche, puis choisissez Save (Enregistrer) pour enregistrer la configuration.
3. Choisissez Instances, ajoutez une autre instance à la couche et démarrez-la.

   L'instance doit être nommée `recipetest2`. Une fois le démarrage terminé, AWS OpsWorks Stacks s'exécutera. `windowstest::default`
4. Une fois l'instance `recipetest2` en ligne, vérifiez que `c:\data` est présent.

> **Note**
>
> Si vous avez attribué des recettes aux événements Setup, Configure ou Deploy, vous pouvez aussi les exécuter manuellement en utilisant une commande de pile (Setup et Configure) ou une commande de déploiement (Deploy) pour déclencher l'événement. Notez que si vous avez plusieurs recettes attribuées à un événement, ces commandes les exécutent toutes.

Exécution d'un PowerShell script Windows

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients

de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Ces exemples supposent que vous avez déjà fait l'exemple Exécution d'une recette sur une instance Windows. Si tel n'est pas le cas, commencez par cet exemple. Il décrit plus particulièrement comment autoriser l'accès RDP à vos instances.

Pour qu'une recette exécute des tâches sur une instance Windows, en particulier des tâches pour lesquelles aucune ressource Chef n'est correspondante, la recette exécute un script Windows. PowerShell Cette section présente les principes de base en expliquant comment utiliser un PowerShell script Windows pour installer une fonctionnalité Windows.

La `powershell_script`ressource exécute des PowerShell applets de commande Windows sur une instance. L'exemple suivant utilise une WindowsFeature applet de commande Install- pour installer un visualiseur XPS sur l'instance.

Voici un bref résumé de la création d'une pile pour cet exemple. Pour de plus amples informations, veuillez consulter Créer une pile.

Création d'une pile

1. Ouvrez la console AWS OpsWorks Stacks et choisissez Add Stack (Ajouter une pile). Spécifiez les paramètres suivants, acceptez les valeurs par défaut pour les autres paramètres et cliquez sur Add Stack (Ajouter une pile).

   - Nom — PowerShellTest
   - Région — Ouest des États-Unis (Oregon)

     Cet exemple fonctionne dans n'importe quelle région, mais nous vous recommandons d'utiliser US West (Oregon) pour les didacticiels.
   - Système d'exploitation par défaut : Microsoft Windows Server 2012 R2
2. Choisissez Add a layer (Ajouter une couche) et ajoutez une couche personnalisée à la pile avec les paramètres suivants.

- Nom — PowerShell
- Nom abrégé : PowerShell

3. Ajoutez une instance 24 h/24 et 7 j/7 à la PowerShell couche avec les paramètres par défaut et démarrez-la.
4. Choisissez Permissions (Autorisations), puis Edit (Modifier), et sélectionnez SSH/RDP et sudo/admin. Vous avez besoin de cette autorisation en plus du groupe de sécurité `AWS-OpsWorks-RDP-Server` pour vous connecter à l'instance en tant qu'utilisateur standard.

Pendant le démarrage de l'instance (cela prend généralement plusieurs minutes), vous pouvez créer le livre de recettes. La recette de cet exemple crée un répertoire de données et est en fait la recette de Exemple 3 : Création de répertoires, modifiée pour Windows.

Pour configurer le livre de recettes

1. Créez un répertoire nommé `powershell` et accédez à celui-ci.
2. Créez un fichier `metadata.rb` avec le contenu suivant et enregistrez-le sur `windowstest`.

```
name "powershell"
version "0.1.0"
```

3. Créez un répertoire `recipes` dans le répertoire `powershell`.
4. Créez un fichier `default.rb` avec la recette suivante et enregistrez-le dans le répertoire `recipes`.

```
Chef::Log.info("******Installing XPS.******")

powershell_script "Install XPS Viewer" do
  code <<-EOH
    Install-WindowsFeature XPS-Viewer
  EOH
  guard_interpreter :powershell_script
  not_if "(Get-WindowsFeature -Name XPS-Viewer).installed"
end
```

- La ressource `powershell_script` exécute une applet de commande pour installer XPS Viewer.

  Cet exemple exécute une seule applet de commande, mais le bloc `code` peut contenir n'importe quel nombre de lignes de commande.
- L'`guard_interpreter`attribut indique à Chef d'utiliser la version 64 bits de Windows PowerShell.
- L'attribut de protection `not_if` veille à ce que Chef n'installe pas la fonctionnalité si elle a déjà été installée.

5. Créez une archive `.zip` du répertoire `powershell`.
6. Téléchargez l'archive dans un compartiment Amazon S3, rendez-la publique et enregistrez l'URL de l'archive. Vous pouvez également utiliser une archive privée, mais une archive publique est suffisante pour cet exemple et légèrement plus facile à utiliser.

   Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Vous pouvez maintenant installer le livre de recettes et exécuter la recette.

Pour exécuter la recette

1. Modifiez la pile pour activer les livres personnalisés et spécifiez les paramètres suivants.

   - Type de référentiel — S3 Archive
   - URL du référentiel : URL de l'archive du livre de recettes que vous avez enregistrée précédemment

   Acceptez les valeurs par défaut pour les autres paramètres, puis choisissez Save (Enregistrer) pour mettre à jour la configuration de la pile.
2. Exécutez la commande de pile Update Custom Cookbooks (Mettre à jour les livres de recettes personnalisées) pour installer la version actuelle de vos livres de recettes personnalisées sur l'instance.

3. Une fois l'opération Update Custom Cookbooks (Mettre à jour les livres de recettes personnalisées) terminée, exécutez la recette en exécutant la commande de pile Execute Recipes (Exécuter les recettes) avec le paramètre Recipes to execute (Recettes à exécuter) défini sur **`powershell::default`**.

> **Note**
>
> Cet exemple utilise Execute Recipes pour des raisons pratiques, mais AWS OpsWorks Stacks exécute généralement vos recettes automatiquement en les affectant à l'événement du cycle de vie approprié. Vous pouvez exécuter ces recettes en déclenchant manuellement l'événement. Vous pouvez utiliser une commande de pile pour déclencher des événements Setup et Configure et une commande de déploiement pour déclencher des événements Deploy et Undeploy.

Une fois que la recette a été exécutée correctement, vous pouvez la vérifier.

Pour vérifier la recette powershell

1. Examinez le journal de Chef. Cliquez sur show (afficher) dans la colonne Log (Journal) de l'instance powershell1. Faites défiler vers le bas pour afficher votre message de journal proche de la fin.

```
...
[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. Utilisez RDP pour vous connecter à l'instance et ouvrez le menu Start (Démarrer). La visionneuse XPS doit être répertoriée avec les Accessoires Windows.

Simulation des attributs de configuration et de déploiement de la pile sur Vagrant

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette rubrique s'applique uniquement aux instances Linux. Test Kitchen n'étant pas encore compatible avec Windows, vous allez exécuter tous les exemples Windows sur des instances AWS OpsWorks Stacks.

AWS OpsWorks Stacks ajoute des attributs de configuration et de déploiement à l'objet nœud pour chaque instance de votre pile pour chaque événement du cycle de vie. Ces attributs fournissent un instantané de la configuration de la pile, y compris la configuration de chaque couche et ses instances en ligne, la configuration de chaque application déployée, etc. Comme ces attributs se trouvent dans l'objet nœud, ils sont accessibles par n'importe quelle recette ; la plupart des recettes pour les instances AWS OpsWorks Stacks utilisent un ou plusieurs de ces attributs.

Une instance exécutée dans une boîte Vagrant n'est pas gérée par AWS OpsWorks Stacks, son objet de nœud n'inclut donc aucune configuration de pile ni aucun attribut de déploiement par défaut. Toutefois, vous pouvez ajouter un ensemble d'attributs approprié à l'environnement de Test Kitchen. Test Kitchen ajoute ensuite les attributs à l'objet nœud de l'instance, et vos recettes peuvent accéder aux attributs de la même manière qu'elles le feraient sur une instance AWS OpsWorks Stacks.

Cette rubrique montre comment obtenir une copie d'attributs de configuration et de déploiement de pile appropriés, installer les attributs sur une instance et y accéder.

> **Note**
>
> Si vous utilisez Test Kitchen pour exécuter des tests sur vos recettes, fauxhai fournit un autre moyen de simuler le JSON de configuration et de déploiement de la pile.

Pour configurer le livre de recettes

1. Créez un sous-répertoire d'`opsworks_cookbooks`, nommé `printjson` et accédez à celui-ci.
2. Initialisez et configurez Test Kitchen, comme décrit dans Exemple 1 : Installation des packages.
3. Ajoutez deux sous-répertoires à `printjson` : `recipes` et `environments`.

Vous pouvez simuler attributs de déploiement et de configuration de la pile en ajoutant un fichier d'attribut à votre livre de recettes avec les définitions appropriées, mais une meilleure approche consiste à utiliser l'environnement de Test Kitchen. Il existe deux approches de base :

- Ajoutez des définitions d'attribut à `.kitchen.yml`.

  Cette approche est particulièrement utile si vous avez quelques attributs. Pour plus d'informations, consultez kitchen.yml.
- Définissez les attributs dans un fichier d'environnement et référencez le fichier dans `.kitchen.yml`.

  Cette approche est généralement préférable pour les attributs de déploiement et de configuration de la pile, car le fichier d'environnement est déjà au format JSON. Vous pouvez obtenir une copie des attributs au format JSON à partir d'une instance AWS OpsWorks Stacks appropriée et simplement la coller. Tous les exemples utilisent un fichier d'environnement.

La façon la plus simple de créer les attributs de configuration et de déploiement d'une pile pour votre livre de recettes consiste à créer une pile configurée de manière appropriée et à copier les attributs obtenus à partir d'une instance au format JSON. Pour veiller à ce que votre fichier d'environnement de Test Kitchen reste gérable, vous pouvez ensuite modifier ce JSON de façon à avoir uniquement les attributs nécessaires pour vos recettes. Les exemples de ce chapitre reposent sur la pile de Mise en route des piles Linux Chef 11, qui est une pile de serveur d'application PHP simple avec un équilibreur de charge, des serveurs d'application PHP et un serveur de base de données MySQL.

Pour créer un JSON de configuration et de déploiement de la pile

1. Créez MyStack comme décrit dansMise en route des piles Linux Chef 11, y compris le déploiement de SimplePHPApp. Si vous préférez, vous pouvez omettre la deuxième instance de PHP App Server appelée dans Étape 4 : Diminution MyStack ; les exemples n'utilisent pas ces attributs.
2. Si vous ne l'avez pas encore fait, lancez l'instance `php-app1`, puis connectez-vous avec SSH.

3. Dans la fenêtre du terminal, exécutez la commande agent cli suivante :

```
sudo opsworks-agent-cli get_json
```

   Cette commande affiche les attributs de configuration et de déploiement de la pile de l'instance les plus récents dans la fenêtre de terminal au format JSON.

4. Copiez le JSON dans un fichier `.json` et enregistrez-le dans un emplacement approprié sur votre poste de travail. Les détails dépendent de votre client SSH. Par exemple, si vous utilisez PuTTY sous Windows, vous pouvez exécuter la commande `Copy All to Clipboard`, qui copie tout le texte de la fenêtre de terminal dans le presse-papiers Windows. Vous pouvez ensuite coller le contenu dans un fichier `.json` et modifier ce dernier pour supprimer le texte superflu.

5. Modifiez le MyStack JSON selon vos besoins. Les attributs de configuration et de déploiement de la pile sont nombreux et les livres de recettes utilisent généralement une petite partie d'entre eux. Pour que votre fichier d'environnement reste gérable, vous pouvez modifier le format JSON afin qu'il conserve la structure d'origine, mais contienne uniquement les attributs réellement utilisés par vos livres de recettes.

   Cet exemple utilise une version fortement modifiée du MyStack JSON qui inclut uniquement deux `['opsworks']['stack']` attributs, `['id]` et`['name']`. Créez une version modifiée du MyStack JSON qui ressemble à ce qui suit :

```
{
  "opsworks": {
    "stack": {
      "name": "MyStack",
      "id": "42dfd151-6766-4f1c-9940-ba79e5220b58",
    },
  },
}
```

Pour obtenir ce JSON dans l'objet de nœud de l'instance, vous devez l'ajouter à un environnement de Test Kitchen.

Pour ajouter des attributs de déploiement et de configuration de la pile à l'environnement de Test Kitchen

1. Créez un fichier d'environnement nommé `test.json` avec le contenu suivant et enregistrez-le dans le dossier `environments` du livre de recettes.

```
{
  "default_attributes": {
    "opsworks" : {
      "stack" : {
        "name" : "MyStack",
        "id" : "42dfd151-6766-4f1c-9940-ba79e5220b58"
      }
    }
  },
  "chef_type" : "environment",
  "json_class" : "Chef::Environment"
}
```

   Le fichier d'environnement comporte les éléments suivants :

   - `default_attributes`— Les attributs par défaut au format JSON.

     Ces attributs sont ajoutés à l'objet de nœud avec le type d'attribut `default`, qui est le type utilisé par tous les attributs JSON de configuration et de déploiement de la pile. Cet exemple utilise la version modifiée du JSON de configuration et de déploiement de la pile présenté précédemment.
   - `chef_type`— Définissez cet élément sur`environment`.
   - `json_class`— Définissez cet élément sur`Chef::Environment`.
2. Modifiez `.kitchen.yml` de façon à définir l'environnement de Test Kitchen comme suit.

```
---
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:
```

Vous définissez l'environnement en ajoutant les éléments suivants à la valeur par défaut `.kitchen.yml` créée par `kitchen init`.

fournisseur

Ajoutez les éléments suivants.

- `name`— Définissez cet élément sur`chef_solo`.

  Pour reproduire plus fidèlement l'environnement AWS OpsWorks Stacks, vous pouvez utiliser le mode local du client Chef au lieu de Chef solo. Le mode local est une option du client Chef qui utilise une version légère du serveur Chef (Chef zéro) exécutée localement sur l'instance plutôt que sur un serveur à distance. Il permet à vos recettes d'utiliser les fonctionnalités du serveur Chef, par exemple les recherches ou les conteneurs de données sans connexion à un serveur à distance.

- `environments_path`— Le sous-répertoire du livre de recettes qui contient le fichier d'environnement, dans cet `./environments` exemple.

suites:provisioner

Ajoutez un élément `solo_rb` avec un élément `environment` défini sur le nom du fichier d'environnement, moins l'extension .json. Cet exemple définit `environment` sur `test`.

3. Créez un fichier nommé `default.rb` avec le contenu suivant et enregistrez-le dans le répertoire `recipes` du livre de recettes.

```
log "Stack name: #{node['opsworks']['stack']['name']}"
log "Stack id: #{node['opsworks']['stack']['id']}"
```

Cette recette consigne simplement les valeurs de la configuration et du déploiement de la pile que vous avez ajoutées à l'environnement. Bien que la recette soit exécutée localement dans Virtual Box, vous référencez ces attributs en utilisant la même syntaxe de nœud que si la recette était exécutée sur une instance AWS OpsWorks Stacks.

4. Exécutez `kitchen converge`. Vous devriez voir quelque chose comme la sortie de journal suivante.

```
...
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

...
```

Utilisation des valeurs des attributs de configuration et de déploiement de la pile

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les recettes ont souvent besoin d'informations sur les applications déployées ou la configuration de la pile. Par exemple, vous pouvez avoir besoin d'une liste d'adresses IP de la pile pour créer un fichier de configuration ou d'un répertoire de déploiement d'une application pour créer un répertoire

des journaux. Au lieu de stocker ces données sur un serveur central, AWS OpsWorks Stacks installe un ensemble d'attributs de configuration et de déploiement de la pile dans l'objet nœud de chaque instance pour chaque événement du cycle de vie. Ces attributs représentent l'état actuel de la pile, dont les applications déployées. Les recettes peuvent ensuite obtenir les données dont elles ont besoin dans l'objet de nœud.

> **Note**
>
> Les applications ont parfois besoin d'informations de l'objet de nœud, par exemple les valeurs d'attribut de déploiement et de configuration de la pile. Toutefois, une application ne peut pas accéder à l'objet de nœud. Pour fournir des données de nœud d'objet à une application, vous pouvez implémenter une recette qui extrait les informations nécessaires dans l'objet de nœud et les place dans un fichier dans un format pratique. L'application peut ensuite lire les données dans le fichier. Pour plus d'informations et pour voir un exemple, consultez Transmission de données aux applications.

Les recettes peuvent obtenir les valeurs des attributs de déploiement et de configuration de la pile dans l'objet de nœud comme suit.

- Directement, à l'aide du nom complet d'un attribut.

  Vous pouvez utiliser cette approche avec n'importe quelle pile Linux, mais pas avec les piles Windows.
- Avec la recherche de Chef, que vous pouvez utiliser pour rechercher les valeurs d'attribut sur l'objet de nœud.

  Vous pouvez utiliser cette approche avec les piles Windows et les piles Linux Chef 11.10.

> **Note**
>
> Avec les piles Linux, vous pouvez utiliser l'interface de ligne de commande de l'agent pour obtenir une copie des attributs de déploiement et de configuration de la pile d'une instance au format JSON. Pour de plus amples informations, veuillez consulter Simulation des attributs de configuration et de déploiement de la pile sur Vagrant.

Rubriques

- Obtention directe des valeurs d'attribut
- Obtention des valeurs d'attribut avec la recherche de Chef

Obtention directe des valeurs d'attribut

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette approche fonctionne uniquement pour les piles Linux.

Simulation des attributs de configuration et de déploiement de la pile sur Vagrant montre comment obtenir les données de déploiement et de configuration de la pile en utilisant la syntaxe de nœud pour référencer directement des attributs spécifiques. C'est parfois la meilleure approche. Toutefois, beaucoup d'attributs sont définis dans des collections ou des listes dont le contenu et le nom peuvent varier d'une pile à l'autre et au fil du temps pour une pile spécifique. Par exemple, l'attribut `deploy` contient une liste d'attributs d'applications, qui sont nommés avec le nom court de l'application. Cette liste, y compris le nom des attributs de l'application, varie généralement d'une pile à l'autre, voire d'un déploiement à l'autre.

Elle est souvent plus utile, et parfois même nécessaire, pour obtenir les données requises en énumérant les attributs d'une liste ou une collection. Par exemple, supposons que vous souhaitiez connaître les adresses IP publiques des instances de votre pile. Ces informations se trouvent dans l'attribut `['opsworks']['layers']`, qui est défini sur une table de hachage contenant un élément pour chacune des couches de la pile, nommées avec le nom court de la couche. Chaque élément de la couche est défini sur une table de hachage contenant des attributs de la couche, dont `['instances']`. Cet élément est à son tour défini sur une autre table de hachage contenant un attribut pour chacune des instances de la couche, nommées avec le nom court de l'instance. Chaque attribut de l'instance est défini sur une autre table de hachage qui contient les attributs de l'instance,

y compris `['ip']`, qui représente l'adresse IP publique. Si vous rencontrez des problèmes pour visualiser tout cela, la procédure suivante inclut un exemple au format JSON.

Cet exemple montre comment obtenir des données à partir du JSON de configuration et de déploiement de la pile pour les couches d'une pile.

Pour configurer le livre de recettes

1. Créez un répertoire dans `opsworks_cookbooks`, nommé `listip` et accédez à celui-ci.
2. Initialisez et configurez Test Kitchen, comme décrit dans Exemple 1 : Installation des packages.
3. Ajoutez deux répertoires à `listip` : `recipes` et `environments`.
4. Créez une version JSON modifiée des attributs de MyStack configuration et de déploiement contenant les attributs pertinents. Elle doit ressembler à ce qui suit.

```
{
  "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. Créez un fichier d'environnement nommé `test.json`, collez le JSON de l'exemple dans `default_attributes` et enregistrez le fichier dans le dossier `environments` du livre de recettes. Le fichier doit ressembler à ce qui suit (à des fins de concision, la plus grande partie du JSON de l'exemple est représentée par des points de suspension).

```
{
  "default_attributes" : {
    "opsworks": {
      "layers": {
        ...
      }
    }
  },
  "chef_type" : "environment",
  "json_class" : "Chef::Environment"
}
```

6. Remplacez le texte dans `.kitchen.yml` par ce qui suit.

```
---
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:
```

Une fois le livre de recettes configuré, vous pouvez utiliser la recette suivante pour enregistrer la couche IDs.

```
node['opsworks']['layers'].each do |layer, layerdata|
  log "#{layerdata['name']} : #{layerdata['id']}"
end
```

La recette énumère les couches de `['opsworks']['layers']` et consigne chaque nom et chaque ID de couche.

Pour exécuter la recette de journalisation des ID de couche

1. Créez un fichier nommé `default.rb` avec l'exemple de recette et enregistrez-le dans le répertoire `recipes`.
2. Exécutez `kitchen converge`.

La partie pertinente de la sortie doit ressembler à ce qui suit.

```
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
```

Pour répertorier les adresses IP des instances, vous aurez besoin d'une boucle imbriquée comme suit.

```
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
```

La boucle interne itère sur les instances de chaque couche et consigne les adresses IP.

Pour exécuter la recette de journalisation des IP de l'instance

1. Remplacez le code de `default.rb` par l'exemple de recette.
2. Exécutez `kitchen converge` pour exécuter la recette.

La partie pertinente de la sortie doit ressembler à ce qui suit.

```
  * 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
```

Lorsque vous avez terminé, exécutez `kitchen destroy` ; la rubrique suivante utilise un nouveau livre de recettes.

> **Note**
>
> Un des motifs les plus fréquents de l'énumération d'une collection JSON de configuration et de déploiement de la pile consiste à obtenir des données pour une application déployée spécifique, par exemple son répertoire de déploiement. Pour obtenir un exemple, consultez Recettes Deploy.

Obtention des valeurs d'attribut avec la recherche de Chef

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette approche est disponible pour les piles Windows et les piles Linux Chef 11.10.

L'obtention des valeurs d'attribut de déploiement et de configuration de la pile directement à partir de l'objet de nœud peut être complexe et ne peut pas être utilisée avec des piles Windows. Une autre approche consiste à utiliser la recherche Chef pour rechercher les attributs intéressants. Si vous connaissez le serveur Chef, vous constaterez que Chef search fonctionne un peu différemment avec AWS OpsWorks Stacks. Comme AWS OpsWorks Stacks utilise chef-client en mode local, Chef search dépend d'une version locale du serveur Chef appelée chef-zero, de sorte que la recherche fonctionne sur les données stockées localement dans l'objet nœud de l'instance plutôt que sur un serveur distant.

D'un point de vue pratique, restreindre la recherche aux données stockées localement n'a généralement pas d'importance, car l'objet nœud d'une instance AWS OpsWorks Stacks inclut la configuration de la pile et les attributs de déploiement. Ils contiennent la plupart, sinon la totalité, des données que les recettes obtiendraient généralement du serveur Chef et portent les mêmes noms. Vous pouvez donc généralement utiliser le code de recherche écrit pour le serveur Chef sur les instances AWS OpsWorks Stacks sans modification. Pour de plus amples informations, veuillez consulter Utilisation de Chef Search.

Vous trouverez ci-dessous la structure de base d'une requête de recherche :

```
result = search(:search_index, "key:pattern")
```

- L'index de recherche spécifie les attributs auxquels la requête s'applique et détermine le type d'objet qui est retourné.
- La clé spécifie le nom de l'attribut.
- Le modèle spécifie les valeurs de l'attribut que vous souhaitez récupérer.

  Vous pouvez exécuter une requête sur des valeurs d'attribut spécifiques ou utiliser des caractères génériques pour rechercher une plage de valeurs.
- Le résultat est une liste d'objets qui correspond à la requête, chacun étant est une table de hachage qui contient plusieurs attributs associés.

  Par exemple, si vous utilisez l'index de recherche `node`, la requête retourne une liste des objets de l'instance, une pour chaque instance qui correspond à la requête. Chaque objet est une table de

hachage qui contient un ensemble d'attributs définissant la configuration de l'instance, par exemple le nom d'hôte et l'adresse IP.

Par exemple, la requête suivante utilise l'index de recherche `node`, c'est-à-dire un index Chef standard qui s'applique aux instances de la pile (ou aux nœuds, selon la terminologie de Chef). Il recherche les instances avec le nom d'hôte `myhost`.

```
result = search(:node, "hostname:myhost")
```

La recherche retourne une liste d'objets de l'instance dont le nom d'hôte est `myhost`. Si vous souhaitez utiliser le système d'exploitation de la première instance, par exemple, il sera représenté par `result[0][:os]`. Si la requête retourne plusieurs objets, vous pouvez les énumérer pour récupérer les informations requises.

Les détails de la méthode d'utilisation de la recherche dans une recette varient selon l'utilisation d'une pile Linux ou Windows. Les rubriques suivantes fournissent des exemples pour les deux types de pile.

Rubriques

- Utilisation de la recherche sur une pile Linux
- Utilisation de la recherche sur une pile Windows

Utilisation de la recherche sur une pile Linux

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cet exemple repose sur une pile Linux avec un seul serveur d'application PHP. Il utilise la fonctionnalité de recherche de Chef afin d'obtenir l'adresse IP publique du serveur et met l'adresse dans un fichier du répertoire `/tmp`. Il récupère essentiellement les mêmes informations de l'objet

de nœud que Obtention directe des valeurs d'attribut , mais le code est beaucoup plus simple et ne dépend pas des détails de la structure d'attribut de la configuration et du déploiement de la pile.

Voici un bref résumé de la création de la pile pour cet exemple. Pour de plus amples informations, veuillez consulter Créer une pile.

> **Note**
>
> Si vous n'avez jamais exécuté de recette personnalisée sur une instance AWS OpsWorks Stacks auparavant, vous devez d'abord suivre l'Exécution d'une recette sur une instance Linuxexemple.

Création d'une pile

1. Ouvrez la console AWS OpsWorks Stacks et cliquez sur Add Stack (Ajouter une pile).
2. Spécifiez les paramètres suivants, acceptez les valeurs par défaut pour les autres paramètres et cliquez sur Add Stack (Ajouter une pile).

   - Nom — SearchJSON
   - Clé SSH par défaut : une paire de EC2 clés Amazon

   Si vous devez créer une paire de EC2 clés Amazon, consultez Amazon EC2 Key Pairs. Notez que la paire de clés doit appartenir à la même région AWS que l'instance. L'exemple utilise la région de l'ouest des États-Unis (Oregon).
3. Cliquez sur Ajouter une couche et ajoutez une couche PHP App Server à la pile avec les paramètres par défaut.
4. Ajoutez une instance 24/7 avec les paramètres par défaut de la couche et démarrez-la.

Pour configurer le livre de recettes

1. Créez un répertoire dans `opsworks_cookbooks`, nommé `searchjson` et accédez à celui-ci.
2. Créez un fichier `metadata.rb` avec le contenu suivant et enregistrez-le sur `opstest`.

```
name "searchjson"
version "0.1.0"
```

3. Créez un répertoire `recipes` dans `searchjson`.
4. Créez un fichier `default.rb` avec la recette suivante et enregistrez-le dans le répertoire `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
```

   Les piles Linux prennent uniquement en charge l'index de recherche `node`. La recette utilise cet index pour obtenir une liste des instances dans la couche `php-app`. Etant donné que la couche est connue pour n'avoir qu'une seule instance, la recette attribue simplement la première à `phpserver`. Si la couche comporte plusieurs instances, vous pouvez les énumérer pour récupérer les informations requises. Chaque élément de la liste est une table de hachage qui contient un ensemble d'attributs d'instance. L'attribut `ip` est défini sur l'adresse IP publique de l'instance. Vous pouvez donc représenter cette adresse dans le code de recette suivant sous la forme `phpserver[:ip]`.

   Après avoir ajouté un message dans le journal de Chef, la recette utilise une ressource `file` pour créer un fichier nommé `ip_addresses`. L'attribut `content` est défini sur une représentation de chaîne de `phpserver[:ip]`. Lorsque Chef crée `ip_addresses`, il ajoute cette chaîne dans le fichier.
5. Créez une `.zip` archive de`opsworks_cookbooks`, téléchargez-la dans un compartiment Amazon S3, rendez-la publique et enregistrez l'URL de l'archive. Pour plus d'informations sur les référentiels de livre de recettes, consultez Référentiels de livres de recettes.

   Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Vous pouvez maintenant installer le livre de recettes et exécuter la recette.

Pour exécuter la recette

1. Modifiez la pile pour activer les livres personnalisés et spécifiez les paramètres suivants.

   - Type de référentiel — Http Archive
   - URL du référentiel : URL de l'archive du livre de recettes que vous avez enregistrée précédemment

   Utilisez les valeurs par défaut pour les autres paramètres, puis cliquez sur Save (Enregistrer) pour mettre à jour la configuration de la pile.
2. Modifiez la configuration de couche personnalisée et attribuez-la `searchjson::default` à l'événement Setup de la couche. AWS OpsWorks Stacks exécutera la recette après le démarrage de l'instance ou si vous déclenchez explicitement l'événement Setup.
3. Exécutez la commande de pile de mise à jour des livres de recettes personnalisés, qui installe la version actuelle de votre référentiel de livres de recettes personnalisé sur les instances de la pile. Si une version antérieure du référentiel est présente, cette commande la remplace.
4. Exécutez la recette en choisissant la commande de pile Setup, ce qui déclenche un événement Setup sur l'instance et exécute `searchjson::default`. Laissez la page de configuration de l'exécution de la commande ouverte.

Une fois que la recette a été exécutée correctement, vous pouvez la vérifier.

Pour vérifier searchjson

1. La première étape consiste à rechercher dans le journal de Chef l'événement Setup le plus récent. Sur la page de configuration de l'exécution de la commande, cliquez sur show (afficher) dans la colonne Log (Journal) de l'instance php-app1 pour afficher le journal. Faites défiler vers le bas pour trouver votre message de journal vers le milieu, comme illustré ci-après.

```
...
[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. Utilisez SSH pour vous connecter à l'instance et affichez le contenu `/tmp`, qui doit inclure un fichier nommé `ip_addresses` contenant l'adresse IP.

Utilisation de la recherche sur une pile Windows

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks propose deux options pour utiliser la recherche sur Windows Stacks.

- L'index de recherche `node`, qui peut être utilisé pour interroger un ensemble d'attributs Chef standard.

  Si vous avez des recettes existantes avec un code de recherche qui les utilise`node`, elles fonctionneront généralement sur les AWS OpsWorks piles Stacks sans modification.
- Un ensemble supplémentaire d'index de recherche qui peut être utilisé pour interroger des ensembles d'attributs propres à AWS OpsWorks Stacks et certains attributs standard.

  Ces index sont détaillés dans Utilisation d'index AWS OpsWorks de recherche spécifiques aux piles sur Windows Stacks.

Nous vous conseillons d'utiliser `node` pour extraire des informations standard, par exemple les noms d'hôte ou les adresses IP. Cette approche permettra à vos recettes d'être conformes aux pratiques standard de Chef. Utilisez les index de recherche AWS OpsWorks Stacks pour récupérer des informations spécifiques à AWS OpsWorks Stacks.

Rubriques

- Utilisation de l'index de recherche du nœud sur les piles Windows

- Utilisation d'index AWS OpsWorks de recherche spécifiques aux piles sur Windows Stacks

Utilisation de l'index de recherche du nœud sur les piles Windows

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cet exemple suppose que vous avez déjà fait l'exemple Exécution d'une recette sur une instance Windows. Si tel n'est pas le cas, commencez par cet exemple. Il décrit plus particulièrement comment autoriser l'accès RDP à vos instances.

Cet exemple repose sur une pile Windows avec une seule couche personnalisée et une instance. Il utilise la recherche de Chef avec l'index de recherche `node` pour obtenir l'adresse IP publique du serveur et met l'adresse dans un fichier du répertoire `C:\tmp`. Voici un bref résumé de la création de la pile pour cet exemple. Pour de plus amples informations, veuillez consulter Créer une pile.

Création d'une pile

1. Ouvrez la console AWS OpsWorks Stacks et choisissez Add Stack (Ajouter une pile).
2. Spécifiez les paramètres suivants, acceptez les valeurs par défaut pour les autres paramètres et choisissez Add Stack (Ajouter une pile).

   - Nom — NodeSearch
   - Région — Ouest des États-Unis (Oregon)

     Cet exemple fonctionne dans n'importe quelle région, mais nous vous recommandons d'utiliser US West (Oregon) pour les didacticiels.
   - Système d'exploitation par défaut : Microsoft Windows Server 2012 R2

3. Choisissez Add a layer (Ajouter une couche) et ajoutez une couche personnalisée à la pile avec les paramètres suivants.

   - Nom — IPTest
   - Nom court — iptest

4. Ajoutez une instance t2.micro 24h/24 et 7j/7 avec les paramètres par défaut à la IPTest couche et démarrez-la. Elle sera nommée iptest1.

   AWS OpsWorks Stacks attribue `AWS-OpsWorks-RDP-Server` automatiquement cette instance, ce qui permet aux utilisateurs autorisés de se connecter à l'instance.

5. Choisissez Permissions (Autorisations), puis Edit (Modifier), et sélectionnez SSH/RDP et sudo/admin. Les utilisateurs standard ont besoin de cette autorisation en plus du groupe de sécurité `AWS-OpsWorks-RDP-Server` pour se connecter à l'instance.

> **Note**
>
> Vous pouvez également vous connecter en tant qu'Administrator, mais en ayant recours à une procédure différente. Pour de plus amples informations, veuillez consulter Connexion avec RDP.

Pour configurer le livre de recettes

1. Créez un répertoire nommé `nodesearch` et accédez à celui-ci.
2. Créez un fichier `metadata.rb` avec le contenu suivant et enregistrez-le sur `opstest`.

```
name "nodesearch"
version "0.1.0"
```

3. Créez un répertoire `recipes` dans `nodesearch`.
4. Créez un fichier `default.rb` avec la recette suivante et enregistrez-le dans le répertoire `recipes`.

```
directory 'C:\tmp' do
  rights :full_control, 'Everyone'
  recursive true
```

```
  action :create
end

windowsserver = search(:node, "hostname:iptest*").first
Chef::Log.info("**********The public IP address is:
 '#{windowsserver[:ipaddress]}'**********")

file 'C:\tmp\addresses.txt' do
  content "#{windowsserver[:ipaddress]}"
  rights :full_control, 'Everyone'
  action :create
end
```

La recette exécute les tâches suivantes :

1. Utilisez une ressource de répertoire pour créer un répertoire `C:\tmp` pour le fichier.

   Pour plus d'informations sur cette ressource, consultez Exemple 3 : Création de répertoires.

2. Utilisez la recherche de Chef avec l'index de recherche `node` pour obtenir une liste de nœuds (instances) avec un nom d'hôte commençant par `iptest`.

   Si vous utilisez le thème par défaut, qui crée des noms d'hôtes en ajoutant des entiers au nom abrégé de la couche, cette requête renverra toutes les instances de la couche. IPTest Pour cet exemple, la couche n'a qu'une seule instance, c'est pourquoi la recette attribue simplement la première à `windowsserver`. Pour plusieurs instances, vous pouvez obtenir la liste complète, puis les énumérer.

3. Ajoute un message avec l'adresse IP au journal de Chef pour cette exécution.

   L'objet `windowsserver` est une table de hachage dont l'attribut `ipaddress` est défini sur l'adresse IP publique de l'instance afin que vous puissiez représenter cette adresse dans le code de recette suivant sous la forme `windowsserver[:ipaddress]`. La recette insère la chaîne correspondante dans le message et l'ajoute dans le journal de Chef.

4. Utilise la ressource `file` pour créer un fichier avec l'adresse IP nommée `C:\tmp\addresses.txt`.

   L'attribut `content` de la ressource spécifie le contenu à ajouter au fichier, l'adresse IP publique dans le cas présent.

5. Créez une archive `.zip` de `nodesearch`, chargez l'archive dans un compartiment S3, rendez l'archive publique, puis enregistrez l'URL de l'archive.

Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Vous pouvez maintenant installer le livre de recettes et exécuter la recette.

Pour installer le livre de recettes et exécuter la recette

1. Modifiez la pile pour activer les livres personnalisés et spécifiez les paramètres suivants.

   - Type de référentiel — S3 Archive
   - URL du référentiel : URL de l'archive du livre de recettes que vous avez enregistrée précédemment

   Acceptez les valeurs par défaut pour les autres paramètres, puis choisissez Save (Enregistrer) pour mettre à jour la configuration de la pile.
2. Exécutez la commande de pile de mise à jour des livres de recettes personnalisés, qui installe la version actuelle de vos livres de recettes personnalisés sur les instances de la pile, dont les instances en ligne. Si une version antérieure de vos livres de recettes est présente, cette commande la remplace.
3. Une fois l'opération Update Custom Cookbooks (Mettre à jour les livres de recettes personnalisées) terminée, exécutez la recette en exécutant la commande de pile Execute Recipes (Exécuter les recettes) avec le paramètre Recipes to execute (Recettes à exécuter) défini sur **`nodesearch::default`**. Cette commande lance une exécution de Chef, avec une liste d'exécution composée de votre recette. Laissez la page execute_recipes ouverte.

Une fois que la recette a été exécutée correctement, vous pouvez la vérifier.

Pour vérifier nodesearch

1. Recherchez dans le journal de Chef l'événement execute_recipes le plus récent. Sur la page Running command execute_recipes (Exécution de la commande execute_recipes), choisissez show (afficher) dans la colonne Log (Journal) de l'instance iptest1 pour afficher le journal. Faites défiler vers le bas pour trouver votre message de journal près de la fin, comme illustré ci-après.

```
...
[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. Utilisez RDP pour vous connecter à l'instance et examinez le contenu de `C:\tmp\addresses.txt`.

Utilisation d'index AWS OpsWorks de recherche spécifiques aux piles sur Windows Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cet exemple suppose que vous avez déjà fait l'exemple Exécution d'une recette sur une instance Windows. Si tel n'est pas le cas, commencez par cet exemple. Il décrit plus particulièrement comment autoriser l'accès RDP à vos instances.

AWS OpsWorks Stacks fournit les index de recherche suivants en plus de : `node`

- `aws_opsworks_stack`— La configuration de la pile.

- `aws_opsworks_layer`— Les configurations des couches de la pile.
- `aws_opsworks_instance`— Les configurations d'instance de la pile.
- `aws_opsworks_app`— Les configurations des applications de la pile.
- `aws_opsworks_user`— Les configurations utilisateur de la pile.
- `aws_opsworks_rds_db_instance`— Informations de connexion pour les instances RDS enregistrées.

Ces index incluent certains attributs Chef standard, mais sont principalement destinés à récupérer des attributs spécifiques à AWS OpsWorks Stacks. Par exemple, `aws_opsworks_instance` inclut un attribut `status` qui fournit l'état de l'instance, par exemple `online`.

> **Note**
>
> La méthode recommandée consiste à utiliser `node` dès que possible pour que vos recettes soient conformes aux pratiques standard de Chef. Pour obtenir un exemple, consultez Utilisation de l'index de recherche du nœud sur les piles Windows.

Cet exemple montre comment utiliser les index AWS OpsWorks Stacks pour récupérer la valeur d'un attribut spécifique à AWS OpsWorks Stacks. Il repose sur une simple pile Windows avec une couche personnalisée qui a une instance. Il utilise la recherche Chef pour obtenir l'ID AWS OpsWorks Stacks de l'instance et place les résultats dans le journal Chef.

Voici un bref résumé de la création d'une pile pour cet exemple. Pour de plus amples informations, veuillez consulter Créer une pile.

Création d'une pile

1. Ouvrez la console AWS OpsWorks Stacks et choisissez + Stack. Spécifiez les paramètres suivants, acceptez les valeurs par défaut pour les autres paramètres et choisissez Add Stack (Ajouter une pile).

   - Nom — IDSearch
   - Région — Ouest des États-Unis (Oregon)

     Cet exemple fonctionne dans n'importe quelle région, mais nous vous recommandons d'utiliser US West (Oregon) pour les didacticiels.

- Système d'exploitation par défaut : Microsoft Windows Server 2012 R2

2. Choisissez Add a layer (Ajouter une couche) et ajoutez une couche personnalisée à la pile avec les paramètres suivants.

   - Nom — IDCheck
   - Nom court — idcheck

3. Ajoutez une instance t2.micro 24h/24 et 7j/7 avec les paramètres par défaut à la IDCheck couche et démarrez-la. Elle sera nommée iptest1.

   AWS OpsWorks Stacks est automatiquement attribué `AWS-OpsWorks-RDP-Server` à cette instance. Activation de l'accès RDPexplique comment ajouter une règle entrante à ce groupe de sécurité qui permet aux utilisateurs autorisés de se connecter à l'instance.

4. Choisissez Permissions (Autorisations), puis Edit (Modifier), et choisissez SSH/RDP et sudo/admin. Les utilisateurs standard ont besoin de cette autorisation en plus du groupe de sécurité `AWS-OpsWorks-RDP-Server` pour se connecter à l'instance.

   > **Note**
   >
   > Vous pouvez également vous connecter en tant qu'Administrator, mais en ayant recours à une procédure différente. Pour de plus amples informations, veuillez consulter Connexion avec RDP.

Pour configurer le livre de recettes

1. Créez un répertoire nommé `idcheck` et accédez à celui-ci.
2. Créez un fichier `metadata.rb` avec le contenu suivant et enregistrez-le sur `opstest`.

```
name "idcheck"
version "0.1.0"
```

3. Créez un répertoire `recipes` dans `idcheck` et ajoutez un fichier `default.rb` dans le répertoire qui contient la recette suivante.

```
windowsserver = search(:aws_opsworks_instance, "hostname:idcheck*").first
```

```
Chef::Log.info("**********The public IP address is:
 '#{windowsserver[:instance_id]}'**********")
```

La recette utilise la fonctionnalité de recherche Chef avec un index de recherche `aws_opsworks_instance` afin d'obtenir les attributs d'instance de chaque instance de la pile avec un nom d'hôte commençant par `idcheck`. Si vous utilisez le thème par défaut, qui crée des noms d'hôtes en ajoutant des entiers au nom abrégé de la couche, cette requête renverra toutes les instances de la couche. IDCheck Pour cet exemple, la couche n'a qu'une seule instance, c'est pourquoi la recette attribue simplement la première à `windowsserver`. Pour plusieurs instances, vous pouvez obtenir la liste complète, puis les énumérer.

La recette profite de la présence d'une seule instance dans la pile avec ce nom d'hôte, c'est pourquoi le premier résultat est le bon. Si la pile comporte plusieurs instances, la recherche sur d'autres attributs peut retourner plusieurs résultats. Pour obtenir une liste d'attributs d'instance, consultez Conteneur de données de l'instance (aws_opsworks_instance).

Les attributs de l'instance sont essentiellement une table de hachage, et l'identifiant AWS OpsWorks Stacks de l'instance est attribué à l'`instance_id`attribut. Vous pouvez donc appeler l'ID. `windowsserver[:instance_id]` La recette insère la chaîne correspondante dans le message et l'ajoute dans le journal de Chef.

4. Créez une `.zip` archive du `ipaddress` livre de recettes, chargez l'archive dans un compartiment Amazon S3 et enregistrez l'URL de l'archive. Pour plus d'informations sur les référentiels de livre de recettes, consultez Référentiels de livres de recettes.

   Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Vous pouvez maintenant installer le livre de recettes et exécuter la recette.

Pour installer le livre de recettes et exécuter la recette

1. Modifiez la pile pour activer les livres personnalisés et spécifiez les paramètres suivants.

   - Type de référentiel — S3 Archive
   - URL du référentiel : URL de l'archive du livre de recettes que vous avez enregistrée précédemment

Acceptez les valeurs par défaut pour les autres paramètres, puis choisissez Save (Enregistrer) pour mettre à jour la configuration de la pile.

2. Exécutez la commande de pile de mise à jour des livres de recettes personnalisés, qui installe la version actuelle de vos livres de recettes personnalisés sur les instances de la pile, dont les instances en ligne. Si une version antérieure de vos livres de recettes est présente, cette commande la remplace.
3. Une fois la mise à jour des livres de recettes personnalisées terminée, exécutez la recette en exécutant la commande de pile Execute Recipes (Exécuter les recettes) avec le paramètre Recipes to execute (Recettes à exécuter) défini sur **`idcheck::default`**. Cette commande lance une exécution de Chef, avec une liste d'exécution composée de votre recette. Laissez la page execute_recipes ouverte.

Une fois que la recette a été exécutée avec succès, vous pouvez la vérifier en recherchant dans le journal de Chef l'événement execute_recipes le plus récent. Sur la page Running command execute_recipes (Exécution de la commande execute_recipes), choisissez show (afficher) dans la colonne Log (Journal) de l'instance iptest1 pour afficher le journal. Faites défiler vers le bas pour trouver votre message de journal près de la fin, comme illustré ci-après.

```
...
[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
...
```

Utilisation d'un livre de recettes externe sur une instance Linux : Berkshelf

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des

questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Berkshelf est disponible uniquement pour les piles Linux de Chef 11.10.

Avant de commencer à implémenter un livre de recettes, consultez Livres de recettes de la communauté Chef, qui contient les livres de recettes qui ont été créés par les membres de la communauté Chef pour un large éventail d'objectifs. Beaucoup de ces livres de recettes peuvent être utilisés avec AWS OpsWorks Stacks sans modification. Vous pourrez donc peut-être en tirer parti pour certaines de vos tâches au lieu d'implémenter tout le code vous-même.

Pour utiliser un livre de recettes externe sur une instance, vous avez besoin de l'installer et de gérer toutes les dépendances. L'approche la plus judicieuse consiste à implémenter un livre de recettes qui prend en charge un gestionnaire de dépendances nommé Berkshelf. Berkshelf fonctionne sur les EC2 instances Amazon, y compris les instances AWS OpsWorks Stacks, mais il est également conçu pour fonctionner avec Test Kitchen et Vagrant. Cependant, l'utilisation sur Vagrant est un peu différente de celle de AWS OpsWorks Stacks. Cette rubrique inclut donc des exemples pour les deux plateformes. Pour plus d'informations sur l'utilisation de Berkshelf, consultez Berkshelf.

Rubriques

- Utilisation de Berkshelf avec Test Kitchen et Vagrant
- Utiliser Berkshelf avec Stacks AWS OpsWorks

Utilisation de Berkshelf avec Test Kitchen et Vagrant

Cet exemple montre comment utiliser Berkshelf pour installer le livre de recettes de mise en route de la communauté et exécuter ses recettes, ce qui installe un fichier texte concis dans votre répertoire de base sur l'instance.

Pour installer Berkshelf et initialiser un livre de recettes

1. Sur votre poste de travail, installez le GEM Berkshelf comme suit.

```
gem install berkshelf
```

En fonction de votre poste de travail, cette commande peut avoir besoin de `sudo`, ou vous pouvez également utiliser un gestionnaire de l'environnement Ruby comme RVM. Pour vérifier que Berkshelf a été installé avec succès, exécutez `berks --version`.

2. Le livre de recettes de cette rubrique est nommé external_cookbook. Vous pouvez utiliser Berkshelf pour créer un livre de recettes initialisé au lieu de l'approche manuelle adoptées pour les rubriques précédentes. Pour cela, accédez au répertoire `opsworks_cookbooks` et exécutez la commande suivante.

   ```
   berks cookbook external_cookbook
   ```

   La commande crée le répertoire `external_cookbook` et plusieurs sous-répertoires Chef et Test Kitchen standard, dont `recipes` et `test`. La commande crée également des versions par défaut d'un certain nombre de fichiers standard, y compris les éléments suivants :

   - `metadata.rb`
   - Fichiers de configuration pour Vagrant, Test Kitchen et Berkshelf
   - Une recette `default.rb` vide dans le répertoire `recipes`

   > **Note**
   >
   > Vous n'avez pas besoin d'exécuter `kitchen init` ; la commande `berks cookbook` gère ces tâches.

3. Exécutez `kitchen converge`. Le livre de recettes nouvellement créé ne fait rien d'intéressant pour l'instant, mais il converge.

> **Note**
>
> Vous pouvez également utiliser `berks init` pour initialiser un livre de recettes existant afin d'utiliser Berkshelf.

Pour utiliser Berkshelf afin de gérer les dépendances externes d'un livre de recettes, le répertoire racine de ce dernier doit comporter un fichier `Berksfile`, c'est-à-dire un fichier de configuration qui spécifie comment Berkshelf doit gérer les dépendances. Lorsque vous avez utilisé `berks`

`cookbook` pour créer le livre de recettes `external_cookbook`, il a créé un fichier `Berksfile` avec le contenu suivant.

```
source "https://supermarket.chef.io"
metadata
```

Ce fichier a les déclarations suivantes :

- `source`— L'URL d'une source de livre de recettes.

  Un Berksfile peut comporter n'importe quel nombre de déclarations `source`, chacune spécifiant une source par défaut pour les livres de recettes dépendants. Si vous ne spécifiez pas explicitement la source d'un livre de recettes, Berkshelf recherche dans les référentiels par défaut un livre de recettes portant le même nom. Le Berksfile par défaut comprend un seul attribut `source` qui spécifie le référentiel du livre de recettes de la communauté. Ce référentiel contient le livre de recettes de mise en route, c'est pourquoi vous pouvez laisser la ligne telle quelle.
- `metadata`— Demande à Berkshelf d'inclure les dépendances du livre de recettes déclarées dans le fichier du livre de recettes. `metadata.rb`

  Vous pouvez également déclarer un livre de recettes dépendant dans le Berksfile en incluant un attribut `cookbook`, comme indiqué plus tard.

Il existe deux façons de déclarer une dépendance de livre de recettes :

- En incluant une déclaration `cookbook` dans le Berksfile.

  C'est l'approche utilisée par AWS OpsWorks Stacks. Par exemple, pour spécifier le livre de recettes de mise en route utilisé dans cet exemple, ajoutez `cookbook "getting-started"` dans le Berksfile. Berkshelf recherchera ensuite dans les référentiels par défaut un livre de recettes portant ce nom. Vous pouvez également utiliser `cookbook` afin de spécifier explicitement une livre de recettes source, voire une version particulière. Pour plus d'informations, consultez Berkshelf.
- En intégrant une déclaration `metadata` dans le Berksfile et en déclarant la dépendance dans `metadata.rb`.

  Cette déclaration fait en sorte que Berkshelf intègre les dépendances de livre de recettes qui sont déclarées dans `metadata.rb`. Par exemple, pour déclarer une dépendance de mise en route,

ajoutez une déclaration `depends 'getting-started'` dans le fichier `metadata.rb` du livre de recettes.

Cet exemple utilise la première approche, par souci de cohérence avec AWS OpsWorks Stacks.

Pour installer le livre de recettes de mise en route

1. Modifiez le Berksfile par défaut pour remplacer la déclaration `metadata` par une déclaration `cookbook` pour `getting-started`. Le contenu doit ressembler à ce qui suit.

```
source "https://supermarket.chef.io"

cookbook 'getting-started'
```

2. Exécutez `berks install` qui télécharge le livre de recettes de mise en route à partir du référentiel du livre de recettes de la communauté et le copie fans le répertoire Berkshelf de votre station de travail, généralement `~/.berkshelf`. Ce répertoire est souvent simplement nommé le Berkshelf. Recherchez dans le répertoire `cookbooks` le répertoire du livre de recettes de mise en route, dont le nom devrait être similaire à `getting-started-0.4.0`.
3. Remplacez `external_cookbook::default` dans la liste d'exécution `.kitchen.yml` par `getting-started::default`. Cet exemple n'exécute aucune recette d'external_cookbook, il utilise simplement le livre de recettes de mise en route. Le fichier `.kitchen.yml` devrait ressembler à ce qui suit.

```
---
driver:
  name: vagrant

provisioner:
  name: chef_solo

platforms:
  - name: ubuntu-12.04

suites:
  - name: default
    run_list:
      - recipe[getting-started::default]
```

```
    attributes:
```

4. Exécutez `kitchen converge`, puis utilisez `kitchen login` pour vous connecter à l'instance. Le répertoire de connexion doit contenir un fichier nommé `chef-getting-started.txt` avec des éléments similaires à ce qui suit :

   ```
   Welcome to Chef!

   This is Chef version 11.12.8.
   Running on ubuntu.
   Version 12.04.
   ```

   Test Kitchen installe les livres de recettes dans le répertoire `/tmp/kitchen/cookbooks` de l'instance. Si vous répertoriez le contenu de ce répertoire, vous verrez deux livres de recettes : external_cookbook et getting-started.
5. Exécutez `kitchen destroy` pour arrêter l'instance. L'exemple suivant utilise une instance AWS OpsWorks Stacks.

Utiliser Berkshelf avec Stacks AWS OpsWorks

AWS OpsWorks Stacks prend en charge en option les stacks Berkshelf for Chef 11.10. Pour utiliser Berkshelf avec votre pile, vous devez effectuer les opérations suivantes.

- Activer Berkshelf pour la pile.

  AWS OpsWorks Stacks gère ensuite les détails de l'installation de Berkshelf sur les instances de la pile.
- Ajouter un Berksfile au répertoire racine du référentiel de livre de recettes.

  Le Berksfile doit contenir les déclarations `source` et `cookbook` pour tous les livres de recettes dépendants.

Lorsque AWS OpsWorks Stacks installe votre référentiel de livres de recettes personnalisé sur une instance, il utilise Berkshelf pour installer les livres de recettes dépendants déclarés dans le Berksfile du référentiel. Pour de plus amples informations, veuillez consulter Utilisation de Berkshelf.

Cet exemple montre comment utiliser Berkshelf pour installer le livre de recettes communautaire Getting-Started sur une instance Stacks. AWS OpsWorks Il installe également une version du

livre de recettes personnalisé createfile, qui crée un fichier dans un répertoire spécifié. Pour plus d'informations sur le fonctionnement de createfile, consultez Installation d'un fichier à partir d'un livre de recettes.

> **Note**
>
> Si c'est la première fois que vous installez un livre de recettes personnalisé sur une pile AWS OpsWorks Stacks, vous devez d'abord suivre l'Exécution d'une recette sur une instance Linuxexemple.

Commencez par créer une pile, comme résumé ci-après. Pour de plus amples informations, veuillez consulter Créer une pile.

Création d'une pile

1. Ouvrez la console AWS OpsWorks Stacks et cliquez sur Add Stack (Ajouter une pile).
2. Spécifiez les paramètres suivants, acceptez les valeurs par défaut pour les autres paramètres et cliquez sur Add Stack (Ajouter une pile).

   - Nom — BerksTest
   - Clé SSH par défaut : une paire de EC2 clés Amazon

   Si vous devez créer une paire de EC2 clés Amazon, consultez Amazon EC2 Key Pairs. Notez que la paire de clés doit appartenir à la même région AWS que l'instance. L'exemple utilise la région USA Ouest (Oregon) par défaut.
3. Cliquez sur Add a layer (Ajouter une couche) et ajoutez une couche personnalisée à la pile avec les paramètres suivants.

   - Nom — BerksTest
   - Nom abrégé — berkstest

   Vous pouvez en fait utiliser n'importe quel type de couche pour cet exemple. Toutefois, l'exemple n'a pas besoin d'un des packages qui sont installés par les autres couches, une couche personnalisée est donc l'approche la plus simple.
4. Ajoutez une instance 24 h/24 et 7 j/7 à la BerksTest couche avec les paramètres par défaut, mais ne la démarrez pas encore.

Avec AWS OpsWorks Stacks, les livres de recettes doivent se trouver dans un dépôt distant avec une structure de répertoire standard. Vous fournissez ensuite les informations de téléchargement à AWS OpsWorks Stacks, qui télécharge automatiquement le référentiel sur chacune des instances de la pile au démarrage. Pour des raisons de simplicité, le dépôt de cet exemple est une archive publique Amazon S3, mais AWS OpsWorks Stacks prend également en charge les archives HTTP, les référentiels Git et les référentiels Subversion. Pour de plus amples informations, veuillez consulter Référentiels de livres de recettes.

Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Pour créer le référentiel des livres de recettes

1. Dans votre répertoire `opsworks_cookbooks`, créez un répertoire nommé `berkstest_cookbooks`. Si vous le souhaitez, vous pouvez aussi créer ce répertoire dans tout emplacement qui vous convient puisque vous le chargerez dans un référentiel.
2. Ajoutez un fichier nommé Berksfile à `berkstest_cookbooks` avec le contenu suivant.

   ```
   source "https://supermarket.chef.io"

   cookbook 'getting-started'
   ```

   Ce fichier déclare la dépendance du livre de recettes de mise en route et fait en sorte que Berkshelf le télécharge à partir du site de livres de recettes de la communauté.
3. Ajoutez un répertoire `createfile` à `berkstest_cookbooks` qui contient les éléments suivants.

   - Un fichier `metadata.rb` avec le contenu suivant.

     ```
     name "createfile"
     version "0.1.0"
     ```

   - Un répertoire `files/default` qui contient un fichier `example_data.json` avec le contenu suivant.

```
{
  "my_name" : "myname",
  "your_name" : "yourname",
  "a_number" : 42,
  "a_boolean" : true
}
```

Le nom et le contenu du fichier sont arbitraires. La recette copie simplement le fichier dans l'emplacement spécifié.

- Un répertoire `recipes` qui contient un fichier `default.rb` avec le code de recette suivant.

```
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
```

Cette recette crée un répertoire `/srv/www/shared` et copie `example_data.json` dans ce répertoire à partir du répertoire `files` du livre de recettes.

4. Créez une `.zip` archive de`berkstest_cookbooks`, téléchargez-la dans un compartiment Amazon S3, rendez-la publique et enregistrez l'URL de l'archive.

Vous pouvez maintenant installer les livres de recettes et exécuter la recette.

Pour installer les livre de recettes et exécuter les recettes

1. Modifiez la pile pour activer les livres personnalisés et spécifiez les paramètres suivants.

   - Type de référentiel — Http Archive

- URL du référentiel : URL de l'archive du livre de recettes que vous avez enregistrée précédemment
- Gérer Berkshelf — Oui

Les deux premiers paramètres fournissent à AWS OpsWorks Stacks les informations dont il a besoin pour télécharger le référentiel de livres de recettes sur vos instances. Le dernier paramètre permet la prise en charge de Berkshelf, qui télécharge le livre de recettes de mise en route dans l'instance. Acceptez les valeurs par défaut pour les autres paramètres, puis cliquez sur Save (Enregistrer) pour mettre à jour la configuration de la pile.

2. Modifiez la BerksTest couche pour ajouter les recettes suivantes à l'événement du cycle de vie de configuration de la couche.
   - `getting-started::default`
   - `createfile::default`
3. Démarrez l'instance. L'événement Setup se produit une fois le démarrage de l'instance terminé. AWS OpsWorks Stacks installe ensuite le référentiel de livres de recettes, utilise Berkshelf pour télécharger le livre de recettes de démarrage et exécute les recettes de configuration et de déploiement de la couche, y compris et. `getting-started::default` `createfile::default`
4. Une fois que l'instance est en ligne, utilisez SSH pour vous connecter. Vous devez voir ce qui suit :
   - `/srv/www/shared` doit contenir `example_data.json`.
   - `/root` doit contenir `chef-getting-started.txt`.

     AWS OpsWorks Stacks exécute les recettes en tant que root. Getting-Started installe donc le fichier dans le `/root` répertoire plutôt que dans votre répertoire personnel.

Utilisation du SDK pour Ruby : téléchargement de fichiers depuis Amazon S3

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des

questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Il existe certaines tâches, telles que l'interaction avec les services AWS, qui ne peuvent pas être effectuées avec les ressources de Chef. Par exemple, il est parfois préférable de stocker des fichiers à distance et de faire en sorte qu'une recette les télécharge sur l'instance. Vous pouvez utiliser la ressource remote_file pour télécharger les fichiers depuis des serveurs distants. Toutefois, si vous souhaitez stocker vos fichiers dans un compartiment Amazon S3, vous ne `remote_file` pouvez télécharger ces fichiers que si l'ACL autorise l'opération.

Les recettes peuvent utiliser le AWS SDK pour Ruby pour accéder à la plupart des services AWS. Cette rubrique explique comment utiliser le SDK pour Ruby afin de télécharger un fichier depuis un compartiment S3.

Note

Pour plus d'informations sur la façon d'utiliser le AWS SDK pour Ruby pour gérer le chiffrement et le déchiffrement, consultez AWS::S3::S3Object. Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Rubriques

- Utilisation du SDK pour Ruby sur une instance Vagrant
- Utilisation du SDK pour Ruby sur AWS OpsWorks une instance Stacks Linux
- Utilisation du SDK pour Ruby sur AWS OpsWorks une instance Stacks Windows

Utilisation du SDK pour Ruby sur une instance Vagrant

Important

Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des

questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette rubrique décrit comment une recette exécutée sur une instance Vagrant peut utiliser le AWS SDK pour Rubypour télécharger un fichier depuis Amazon S3. Avant de commencer, vous devez disposer d'un ensemble d' AWS informations d'identification (une clé d'accès et une clé d'accès secrète) qui permettent à la recette d'accéder à Amazon S3.

Important

Nous vous déconseillons vivement d'utiliser les informations d'identification du compte racine dans cette optique. Créez plutôt un utilisateur doté d'une politique appropriée et fournissez ces informations d'identification à la recette.

Veillez à ne pas placer les informations d'identification, même les informations d'identification utilisateur IAM, dans un emplacement accessible au public, par exemple en téléchargeant un fichier contenant les informations d'identification dans un référentiel public ou Bitbucket. GitHub Vous risqueriez d'exposer vos informations d'identification et de compromettre la sécurité de votre compte.

Les recettes exécutées sur une EC2 instance EC2 Amazon peuvent utiliser une approche encore meilleure, un rôle IAM, comme décrit dansUtilisation du SDK pour Ruby sur AWS OpsWorks une instance Stacks Linux.

Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Si vous n'avez pas encore d'utilisateur approprié, vous pouvez en créer un comme suit. Pour plus d'informations, consultez Qu'est-ce qu'IAM ?

Warning

Les utilisateurs IAM disposent d'informations d'identification à long terme, ce qui présente un risque de sécurité. Pour atténuer ce risque, nous vous recommandons de ne fournir à ces utilisateurs que les autorisations dont ils ont besoin pour effectuer la tâche et de supprimer ces autorisations lorsqu'elles ne sont plus nécessaires.

Pour créer un utilisateur IAM

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à https://console.aws.amazon.com/iam/l'adresse.
2. Dans le volet de navigation, sélectionnez Utilisateurs et, si nécessaire, choisissez Ajouter des utilisateurs pour créer un nouvel utilisateur administratif.
3. Sur la page Définir les autorisations, choisissez Joindre directement les politiques.
4. Tapez **S3** dans le champ de recherche des politiques d'autorisation pour afficher les politiques Amazon S3.

   Choisissez Amazon S3 ReadOnlyAccess. Si vous préférez, vous pouvez définir une politique qui accorde des autorisations plus étendues, comme AmazonS3 FullAccess, mais la pratique standard consiste à n'accorder que les autorisations requises. Dans ce cas, la recette ne fera que télécharger un fichier, un accès en lecture seule est donc suffisant.
5. Choisissez Suivant.
6. Choisissez Créer un utilisateur
7. Créez ensuite des clés d'accès pour votre utilisateur. Pour de plus amples informations sur la création de clés d'accès, veuillez consulter Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Vous devez ensuite fournir un fichier à télécharger. Cet exemple suppose que vous placiez un fichier nommé `myfile.txt` dans un compartiment S3 nouvellement créé nommé `cookbook_bucket`.

Pour fournir un fichier à télécharger

1. Créez un fichier nommé `myfile.txt` avec le texte suivant et enregistrez-le dans un emplacement approprié sur votre ordinateur.

   ```
   This is the file that you just downloaded from Amazon S3.
   ```
2. Sur la console Amazon S3, créez un compartiment nommé `cookbook_bucket` dans la région Standard et chargez-le `myfile.txt` dans le compartiment.

Configurez le livre de recettes comme suit.

Pour configurer le livre de recettes

1. Créez un répertoire dans `opsworks_cookbooks`, nommé `s3bucket` et accédez à celui-ci.
2. Initialisez et configurez Test Kitchen, comme décrit dans Exemple 1 : Installation des packages.
3. Remplacez le texte dans `.kitchen.yml` par ce qui suit.

```
---
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. Ajoutez deux répertoires à `s3bucket` : `recipes` et `environments`.
5. Créez un fichier d'environnement nommé `test.json` dans la `default_attributes` section suivante, en remplaçant les `secret_key` valeurs `access_key` et par les clés correspondantes pour votre utilisateur. Enregistrez le fichier dans le dossier `environments` du livre de recettes.

```
{
  "default_attributes" : {
    "cookbooks_101" : {
      "access_key": "AKIAIOSFODNN7EXAMPLE",
      "secret_key" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
    }
  },
  "chef_type" : "environment",
```

```
  "json_class" : "Chef::Environment"
}
```

Vous pouvez saisir vos informations d'identification de plusieurs façons pour une recette exécutée sur une instance. Le plus important est de limiter les risques d'exposition accidentelle des clés et de compromission de la sécurité de votre compte. C'est pourquoi il n'est pas recommandé d'utiliser des valeurs de clé explicites dans votre code. Dans le cas présent, nous mettons les valeurs de clé dans l'objet de nœud, ce qui permet à la recette de les référencer en utilisant la syntaxe du nœud au lieu d'exposer les valeurs littérales. Vous devez disposer de privilèges racine pour accéder à l'objet de nœud, ce qui limite les risques d'exposition des clés. Pour plus d'informations, consultez Bonnes pratiques en matière de gestion des clés d'accès AWS.

> **Note**
>
> Notez que l'exemple utilise les attributs imbriqués, avec `cookbooks_101` comme premier élément. Cette pratique limite les risques de conflit de noms s'il y a d'autres attributs `access_key` ou `secret_key` dans l'objet de nœud.

La recette suivante télécharge `myfile.text` à partir du compartiment `cookbook_bucket`.

```
gem_package "aws-sdk ~> 3" do
  action :install
end

ruby_block "download-object" do
  block do
    require 'aws-sdk'

    s3 = Aws::S3::Client.new(
          :access_key_id => "#{node['cookbooks_101']['access_key']}",
          :secret_access_key => "#{node['cookbooks_101']['secret_key']}")

    myfile = s3.bucket['cookbook_bucket'].objects['myfile.txt']
    Dir.chdir("/tmp")
    File.open("myfile.txt", "w") do |f|
      f.write(myfile.read)
      f.close
    end
```

```
  end
  action :run
end
```

La première partie de la recette installe le SDK pour Ruby, qui est un package gem. La ressource gem_package installe des gems qui seront utilisées par les recettes ou par d'autres applications.

> **Note**
>
> Votre instance a généralement deux instances Ruby correspondant la plupart du temps à des versions différentes. La première est une instance dédiée qui est utilisée par le client Chef. La deuxième est utilisée par les applications et les recettes exécutées sur l'instance. Il est important de comprendre cette distinction lors de l'installation des packages de gems, car deux ressources permettent d'installer des gems, gem_package et chef_gem. Si des applications ou des recettes utilisent le package gem, installez-le avec `gem_package` ; `chef_gem` est uniquement destiné aux packages GEM utilisés par le client Chef.

Le reste de la recette est une ressource ruby_block qui contient le code Ruby destiné à télécharger le fichier. Vous pourriez penser que dans la mesure où une recette est une application Ruby, vous pouvez placer le code dans la recette directement. Toutefois, une exécution de Chef compile tout ce code avant l'exécution de n'importe quelle ressource. Si vous mettez l'exemple de code directement dans la recette, Ruby tente de résoudre l'instruction `require 'aws-sdk'` avant d'exécuter la ressource `gem_package`. Le SDK pour Ruby n'étant pas encore installé, la compilation échouera.

Le code d'une ressource `ruby_block` n'est pas compilé tant que cette ressource n'a pas été exécutée. Dans cet exemple, la `ruby_block` ressource est exécutée une fois qu'elle a fini d'installer le SDK pour Ruby, de sorte que le code s'exécute correctement. `gem_package`

Le code du `ruby_block` fonctionne comme suit.

1. Crée un nouvel objet `Aws::S3`, qui fournit l'interface de service.

   Les clés d'accès et secrètes sont spécifiées en référençant les valeurs stockées dans l'objet de nœud.
2. Appelle l'association `bucket.objects` de l'objet `S3`, qui retourne un objet `Aws::S3::Object` nommé `myfile` qui représente `myfile.txt`.
3. Utilise `Dir.chdir` pour définir le répertoire de travail sur `/tmp`.

4. Ouvre un fichier nommé `myfile.txt`, écrit le contenu de `myfile` dans le fichier, puis ferme le fichier.

Pour exécuter la recette

1. Créez un fichier nommé `default.rb` avec l'exemple de recette et enregistrez-le dans le répertoire `recipes`.
2. Exécutez `kitchen converge`.
3. Exécutez `kitchen login`, puis connectez-vous à l'instance et exécutez `ls /tmp`. Vous devriez voir `myfile.txt`, ainsi que plusieurs répertoires et fichiers Test Kitchen.

```
vagrant@s3bucket-ubuntu-1204:~$ ls /tmp
install.sh  kitchen  myfile.txt  stderr
```

   Vous pouvez également exécuter `cat /tmp/myfile.txt` afin de vérifier que le contenu du fichier est correct.

Une fois que vous avez terminé, exécutez `kitchen destroy` pour résilier l'instance.

Utilisation du SDK pour Ruby sur AWS OpsWorks une instance Stacks Linux

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette rubrique décrit comment utiliser le SDK pour Ruby sur AWS OpsWorks une instance Stacks Linux pour télécharger un fichier depuis un compartiment Amazon S3. AWS OpsWorks Stacks installe automatiquement le SDK pour Ruby sur chaque instance Linux. Toutefois, lorsque vous créez l'objet client d'un service, vous devez fournir un ensemble approprié d'informations d'identification AWS `AWS::S3.new` ou l'équivalent pour d'autres services.

Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Utilisation du SDK pour Ruby sur une instance Vagrant montre comment atténuer le risque d'exposition de vos informations d'identification en stockant ces dernières dans l'objet de nœud et en référençant les attributs dans votre code de recette. Lorsque vous exécutez des recettes sur une EC2 instance Amazon, vous disposez d'une option encore meilleure : un rôle IAM.

Un rôle IAM fonctionne de la même manière qu'un utilisateur IAM. Il dispose d'une stratégie attachée qui accorde des autorisations d'utilisation des différents services AWS. Cependant, vous attribuez un rôle à une EC2 instance Amazon plutôt qu'à un individu. Les applications exécutées sur cette instance peuvent ensuite acquérir les autorisations accordées par la stratégie attachée. Avec un rôle, les informations d'identification n'apparaissent jamais dans votre code, même indirectement. Cette rubrique décrit comment utiliser un rôle IAM pour exécuter la recette depuis Utilisation du SDK pour Ruby sur une instance Vagrant une EC2 instance Amazon.

Vous pouvez exécuter cette recette avec Test Kitchen à l'aide du pilote kitchen-ec2, comme décrit dans Exemple 9 : utilisation des EC2 instances Amazon. Cependant, l'installation du SDK pour Ruby sur les instances EC2 Amazon est quelque peu compliquée et ne doit pas vous inquiéter AWS OpsWorks pour Stacks. Le SDK pour Ruby est installé par défaut sur toutes les instances AWS OpsWorks Stacks Linux. Pour des raisons de simplicité, l'exemple utilise donc une instance AWS OpsWorks Stacks.

La première étape consiste à configurer le rôle IAM. Cet exemple adopte l'approche la plus simple, qui consiste à utiliser le EC2 rôle Amazon créé par AWS OpsWorks Stacks lorsque vous créez votre première pile. Il est nommé `aws-opsworks-ec2-role`. Cependant, AWS OpsWorks Stacks n'attache pas de politique à ce rôle. Par conséquent, par défaut, il n'accorde aucune autorisation.

Vous devez associer la `AmazonS3ReadOnlyAccess` politique au `aws-opsworks-ec2-role` rôle pour accorder les autorisations appropriées. Pour plus d'informations sur la façon d'associer une politique à un rôle, consultez la section Ajout d'autorisations d'identité IAM (console) dans le guide de l'utilisateur IAM.

Vous spécifiez le rôle lorsque vous créez ou mettez à jour une pile. Configurez une pile avec une couche personnalisée, comme décrit dans Exécution d'une recette sur une instance Linux, avec un ajout. Sur la page Ajouter une pile, vérifiez que le profil d'instance IAM par défaut est défini sur aws-

opsworks-ec2 rôles. AWS OpsWorks Stacks attribuera ensuite ce rôle à toutes les instances de la pile.

La procédure de configuration du livre de recettes est similaire à celle utilisée par Exécution d'une recette sur une instance Linux. Voici un bref résumé ; reportez-vous à cet exemple pour plus de détails.

Pour configurer le livre de recettes

1. Créez un répertoire nommé `s3bucket_ops` et accédez à celui-ci.
2. Créez un fichier `metadata.rb` avec le contenu suivant et enregistrez-le sur `s3bucket_ops`.

```
name "s3bucket_ops"
version "0.1.0"
```

3. Créez un répertoire `recipes` dans `s3bucket_ops`.
4. Créez un fichier `default.rb` avec la recette suivante et enregistrez-le dans le répertoire `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. Créez une `.zip` archive `s3bucket_ops` et chargez-la dans un compartiment Amazon S3. Pour simplifier, rendez l'archive publique, puis enregistrez l'URL de l'archive pour une utilisation

ultérieure. Vous pouvez également stocker vos livres de recettes dans une archive privée Amazon S3 ou dans plusieurs autres types de référentiels. Pour de plus amples informations, veuillez consulter Référentiels de livres de recettes.

Cette recette est similaire celle utilisée par l'exemple précédent, avec les exceptions suivantes.

- AWS OpsWorks Stacks ayant déjà installé le SDK pour Ruby, `chef_gem` la ressource a été supprimée.
- La recette ne passe aucune information d'identification à `AWS::S3.new`.

  Les informations d'identification sont automatiquement attribuées à l'application en fonction du rôle de l'instance.
- La recette utilise `Chef::Log.info` pour ajouter un message dans le journal de Chef.

Créez une pile pour cet exemple, comme suit. Vous pouvez aussi utiliser une pile Windows existante. Il vous suffit de mettre à jour les livres de recettes, comme décrit plus tard.

Pour créer une pile

1. Ouvrez la console AWS OpsWorks Stacks et cliquez sur Add Stack (Ajouter une pile).
2. Spécifiez les paramètres suivants, acceptez les valeurs par défaut pour les autres paramètres et cliquez sur Add Stack (Ajouter une pile).

   - Nom — RubySDK
   - Clé SSH par défaut : une paire de EC2 clés Amazon

   Si vous devez créer une paire de EC2 clés Amazon, consultez Amazon EC2 Key Pairs. Notez que la paire de clés doit appartenir à la même région AWS que l'instance. L'exemple utilise la région USA Ouest (Oregon) par défaut.
3. Cliquez sur Add a layer (Ajouter une couche) et ajoutez une couche personnalisée à la pile avec les paramètres suivants.

   - Nom — S3Download
   - Nom court — s3download

N'importe quel type de couche peut être utilisé pour les piles Linux, mais l'exemple n'a pas besoin des packages installés par les autres types de couches, c'est pourquoi une couche personnalisée constitue l'approche la plus simple.

4. Ajoutez une instance 24/7 avec les paramètres par défaut de la couche et démarrez-la.

Vous pouvez désormais installer et exécuter la recette.

Pour exécuter la recette

1. Modifiez la pile pour activer les livres personnalisés et spécifiez les paramètres suivants.

   - Type de référentiel — Http Archive
   - URL du référentiel : URL de l'archive du livre de recettes que vous avez enregistrée précédemment.

   Utilisez les valeurs par défaut pour les autres paramètres, puis cliquez sur Save (Enregistrer) pour mettre à jour la configuration de la pile.
2. Exécutez la commande de pile de mise à jour des livres de recettes personnalisés, qui installe la version actuelle de vos livres de recettes personnalisés sur les instances de la pile. Si une version antérieure de vos livres de recettes est présente, cette commande la remplace.
3. Exécutez la recette à l'aide de la commande de pile Execute Recipes (Exécuter les recettes) après avoir défini Recipes to execute (Recettes à exécuter) sur **`s3bucket_ops::default`**. Cette commande lance une exécution de Chef, avec une liste d'exécution composée de `s3bucket_ops::default`.

   > **Note**
   >
   > Vous demandez généralement à AWS OpsWorks Stacks d'exécuter automatiquement vos recettes en les affectant à l'événement du cycle de vie approprié. Vous pouvez exécuter ces recettes en déclenchant manuellement l'événement. Vous pouvez utiliser une commande de pile pour déclencher des événements Setup et Configure et une commande de déploiement pour déclencher des événements Deploy et Undeploy.

Une fois que la recette a été exécutée correctement, vous pouvez la vérifier.

Pour vérifier s3bucket_ops

1. La première étape consiste à examiner le journal de Chef. Votre pile doit avoir une seule instance nommée opstest1. Sur la page Instances, cliquez sur show (afficher) dans la colonne Log (Journal) de l'instance pour afficher le journal de Chef. Faites défiler vers le bas et recherchez votre message de journal près de la fin.

   ```
   ...
   [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. Utilisez SSH pour vous connecter à l'instance et affichez la liste du contenu de `/tmp`.

Utilisation du SDK pour Ruby sur AWS OpsWorks une instance Stacks Windows

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cet exemple suppose que vous avez déjà fait l'exemple Exécution d'une recette sur une instance Windows. Si tel n'est pas le cas, commencez par cet exemple. Il décrit plus particulièrement comment autoriser l'accès RDP à vos instances.
> Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3

Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Cette rubrique décrit comment utiliser AWS SDK pour Rubyl'instance AWS OpsWorks Stacks Windows pour télécharger un fichier depuis un compartiment S3.

Si une application Ruby a besoin d'accéder à une ressource AWS, vous devez lui fournir un ensemble d'informations d'identification AWS avec les autorisations appropriées. Pour les recettes, la meilleure option pour fournir des informations d'identification AWS est d'utiliser un rôle AWS Identity and Access Management (IAM). Un rôle IAM fonctionne de la même manière qu'un utilisateur IAM : il est associé à une politique qui accorde des autorisations pour utiliser les différents AWS services. Toutefois, vous attribuez un rôle à une instance Amazon Elastic Compute Cloud (Amazon EC2) plutôt qu'à un individu. Les applications exécutées sur cette instance peuvent ensuite acquérir les autorisations accordées par la stratégie attachée. Avec un rôle, les informations d'identification n'apparaissent jamais dans votre code, même indirectement.

La première étape consiste à configurer le rôle IAM. Cet exemple adopte l'approche la plus simple, qui consiste à utiliser le EC2 rôle Amazon créé par AWS OpsWorks Stacks lorsque vous créez votre première pile. Il est nommé `aws-opsworks-ec2-role`. Cependant, AWS OpsWorks Stacks n'attache pas de politique à ce rôle. Par conséquent, par défaut, il n'accorde aucune autorisation.

Vous devez associer la `AmazonS3ReadOnlyAccess` politique au `aws-opsworks-ec2-role` rôle pour accorder les autorisations appropriées. Pour plus d'informations sur la façon d'associer une politique à un rôle, consultez la section Ajout d'autorisations d'identité IAM (console) dans le guide de l'utilisateur IAM.

Vous spécifiez le rôle lorsque vous créez ou mettez à jour une pile. Configurez une pile avec une couche personnalisée, comme décrit dans Exécution d'une recette sur une instance Windows, avec un ajout. Sur la page Ajouter une pile, vérifiez que le profil d'instance IAM par défaut est défini sur aws-opsworks-ec2 rôles. AWS OpsWorks Stacks attribuera ensuite ce rôle à toutes les instances de la pile.

La procédure de configuration du livre de recettes est similaire à celle utilisée par Exécution d'une recette sur une instance Linux. Voici un bref résumé ; reportez-vous à cet exemple pour plus de détails.

Pour configurer le livre de recettes

1. Créez un répertoire nommé `s3bucket_ops` et accédez à celui-ci.

2. Créez un fichier `metadata.rb` avec le contenu suivant et enregistrez-le sur `s3bucket_ops`.

```
name "s3download"
version "0.1.0"
```

3. Créez un répertoire `recipes` dans `s3download`.
4. Créez un fichier `default.rb` avec la recette suivante et enregistrez-le dans le répertoire `recipes`. *`windows-cookbooks`*Remplacez-le par le nom du compartiment S3 que vous utiliserez pour stocker le fichier à télécharger.

```
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. Créez une archive `.zip` de `s3download` et chargez le fichier dans le compartiment S3. Rendez le fichier public et enregistrez l'URL pour une utilisation ultérieure.
6. Créez un fichier texte nommé `myfile.txt` et chargez-le dans un compartiment S3. C'est le fichier qui sera téléchargé par votre recette, c'est pourquoi vous pouvez utiliser n'importe quel compartiment approprié.

La recette exécute les tâches suivantes.

1 : Installez le SDK pour Ruby v2.

L'exemple utilise le SDK pour Ruby pour télécharger l'objet. Cependant, AWS OpsWorks Stacks n'installe pas ce SDK sur les instances Windows. La première partie de la recette utilise donc une `chef_gem`ressource pour gérer cette tâche. Vous utilisez cette ressource pour installer des gems qui seront utilisés par Chef, ce qui inclut les recettes.

2 : Téléchargement du fichier.

La troisième partie de la recette utilise une `ruby_block`ressource pour exécuter le code SDK for Ruby v2 à `myfile.txt` télécharger depuis un compartiment S3 *`windows-cookbooks`* nommé dans le répertoire `/chef` de l'instance. Remplacez *`windows-cookbooks`* par le nom du compartiment qui contient `myfile.txt`.

> **Note**
>
> Une recette est une application Ruby, ce qui vous permet de mettre le code Ruby dans le corps de la recette ; il ne doit pas être dans une ressource `ruby_block`. Toutefois, Chef exécute le code Ruby dans le corps de la recette en premier, puis chaque ressource, dans l'ordre. Dans cet exemple, si vous insérez le code de téléchargement dans le corps de la recette, il échouera car il dépend du SDK pour Ruby et `chef_gem` la ressource qui installe le SDK n'a pas encore été exécutée. Le code de la `ruby_block` ressource s'exécute lorsque la ressource s'exécute, et cela se produit une fois que la `chef_gem` ressource a installé le SDK pour Ruby.

Créez une pile pour cet exemple, comme suit. Vous pouvez aussi utiliser une pile Windows existante. Il vous suffit de mettre à jour les livres de recettes, comme décrit plus tard.

Création d'une pile

1. Ouvrez la console AWS OpsWorks Stacks et choisissez Add Stack (Ajouter une pile). Spécifiez les paramètres suivants, acceptez les valeurs par défaut pour les autres paramètres et choisissez Add Stack (Ajouter une pile).

   - Nom — S3Download
   - Région — Ouest des États-Unis (Oregon)

Cet exemple fonctionne dans n'importe quelle région, mais nous vous recommandons d'utiliser US West (Oregon) pour les didacticiels.

- Système d'exploitation par défaut : Microsoft Windows Server 2012 R2

2. Choisissez Add a layer (Ajouter une couche) et ajoutez une couche personnalisée à la pile avec les paramètres suivants.

   - Nom — S3Download
   - Nom court — s3download

3. Ajoutez une instance 24/7 avec les paramètres par défaut de la couche S3Download et démarrez-la.

Vous pouvez désormais installer et exécuter la recette.

Pour exécuter la recette

1. Modifiez la pile pour activer les livres personnalisés et spécifiez les paramètres suivants.

   - Type de référentiel : archive S3.
   - URL du référentiel : URL de l'archive du livre de recettes que vous avez enregistrée précédemment.

   Acceptez les valeurs par défaut pour les autres paramètres, puis choisissez Save (Enregistrer) pour mettre à jour la configuration de la pile.

2. Exécutez la commande de pile Update Custom Cookbooks (Mettre à jour les livres de recettes personnalisées), qui installe la dernière version de votre livre de recettes personnalisé sur les instances en ligne de la pile. Si une version antérieure de vos livres de recettes est présente, cette commande la remplace.
3. Exécutez la recette à l'aide de la commande de pile Execute Recipes (Exécuter les recettes) après avoir défini Recipes to execute (Recettes à exécuter) sur **`s3download::default`**. Cette commande lance une exécution de Chef, avec une liste d'exécution composée de `s3download::default`.

> **Note**
>
> Vous demandez généralement à AWS OpsWorks Stacks d'exécuter automatiquement vos recettes en les affectant à l'événement du cycle de vie approprié. Vous pouvez également exécuter ces recettes en déclenchant manuellement l'événement. Vous pouvez utiliser une commande de pile pour déclencher des événements Setup et Configure et une commande de déploiement pour déclencher des événements Deploy et Undeploy.

Une fois que la recette a été exécutée correctement, vous pouvez la vérifier.

Pour vérifier s3download

1. La première étape consiste à examiner le journal de Chef. Votre pile doit avoir une seule instance nommée s3download1. Sur la page Instances, choisissez show (afficher) dans la colonne Log (Journal) de l'instance pour afficher le journal de Chef. Faites défiler vers le bas pour rechercher votre message de journal près de la fin.

```
...
[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. Utilisez RDP pour vous connecter à l'instance et examinez le contenu de `c:\chef`.

Installation de logiciels Windows

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients

de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Ces exemples supposent que vous avez déjà fait l'exemple Exécution d'une recette sur une instance Windows. Si tel n'est pas le cas, commencez par cet exemple. Il décrit plus particulièrement comment autoriser l'accès RDP à vos instances.

Les instances Windows démarrent avec Windows Server 2012 R2 Standard, c'est pourquoi vous devez généralement installer un logiciel. Les détails dépendent du type de logiciel.

- Les fonctionnalités Windows sont des composants système optionnels, notamment les frameworks .NET et Internet Information Services (IIS), que vous pouvez télécharger sur votre instance.
- Des logiciels tiers sont généralement fournis dans un package d'installation, par exemple un fichier MSI, que vous devez télécharger sur l'instance, puis exécuter….

  Certains logiciels Microsoft sont également fournis dans un package d'installation.

Cette section décrit comment implémenter des livres de recettes pour installer les packages et les fonctionnalités de Windows. Elle présente également le livre de recettes Windows de Chef, qui contient des ressources et des fonctions d'assistance destinées à simplifier l'implémentation des recettes pour les instances Windows.

Rubriques

- Installation d'une fonctionnalité de Windows : IIS
- Installation d'un package sur une instance Windows

Installation d'une fonctionnalité de Windows : IIS

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les fonctionnalités Windows sont un ensemble de composants système optionnels, notamment les frameworks .NET et Internet Information Services (IIS). Cette rubrique décrit comment implémenter un livre de recettes pour installer une fonctionnalité couramment utilisée, les services Internet (IIS).

> **Note**
>
> Installation d'un package montre comment installer un logiciel fourni dans un package d'installation, par exemple un fichier MSI, que vous devez télécharger sur l'instance et exécuter. Livres de recettes IIS

Exécution d'une recette sur une instance Windows montre comment utiliser une ressource `powershell_script` pour installer une fonctionnalité de Windows. Cet exemple montre une approche alternative : utilisez la `windows_feature` ressource du livre de recettes Chef Windows. Ce livre contient un ensemble de ressources qui utilisent la maintenance et le déploiement des images de déploiement afin d'effectuer diverses tâches sous Windows, dont l'installation des fonctionnalités.

> **Note**
>
> Chef dispose également d'un livre de recettes IIS, que vous pouvez utiliser pour gérer IIS. Pour plus d'informations, consultez Livre de recettes IIS.

Pour configurer le livre de recettes

1. Accédez au GitHub référentiel de livres de recettes Windows et téléchargez le livre de `windows` recettes.

Cet exemple suppose que vous téléchargiez le référentiel `windows` comme fichier .zip, mais vous pouvez également cloner le référentiel si vous le souhaitez.

2. Accédez au GitHub référentiel de livres de recettes chef_handler et téléchargez le livre de recettes. `chef-handler`

   Le livre de recettes `windows` dépend de `chef_handler` ; vous ne l'utiliserez pas directement. Cet exemple suppose que vous téléchargiez le référentiel `chef_handler` comme fichier .zip, mais vous pouvez également cloner le référentiel si vous le souhaitez.

3. Procédez à l'extraction des livres de recettes `windows` et `chef_handler` et transférez-les dans les répertoires de votre répertoire de livres de recettes nommés `windows` et `chef_handler`, respectivement.

4. Créez un répertoire dans votre répertoire de livres de recettes nommé `install-iis` et accédez à celui-ci.

5. Ajoutez un fichier `metadata.rb` à `install-iis` avec le contenu suivant.

   ```
   name "install-iis"
   version "0.1.0"

   depends "windows"
   ```

   La directive `depends` vous permet d'utiliser les ressources des livres de recettes `windows` dans vos recettes.

6. Ajoutez un répertoire `recipes` à `install-iis` et ajoutez un fichier nommé `default.rb` dans ce répertoire qui contient le code de recette suivant.

   ```
   %w{ IIS-WebServerRole IIS-WebServer }.each do |feature|
     windows_feature feature do
       action :install
     end
   end

   service 'w3svc' do
     action [:start, :enable]
   end
   ```

La recette utilise la ressource `windows` du livre de recettes `windows_feature` pour installer les éléments suivants :

1. Le rôle de serveur web IIS.
2. Le serveur web IIS.

La recette utilise ensuite une ressource `service` pour démarrer et activer le service IIS (W3SVC).

> **Note**
>
> Pour obtenir la liste complète des fonctionnalités Windows disponibles, utilisez RDP pour vous connecter à l'instance, ouvrez une fenêtre d'invite de commande et exécutez la commande suivante. Notez que la liste est relativement longue.
>
> ```
> dism /online /Get-Features
> ```

7. Créez une archive `.zip` qui contient les livres de recettes `install-iis`, `chef_handler` et `windows`, puis chargez l'archive dans un compartiment S3. Rendez l'archive publique et enregistrez l'URL pour une utilisation ultérieure. Cet exemple suppose que l'archive se nomme `install-iis.zip`. Pour de plus amples informations, veuillez consulter Référentiels de livres de recettes.

   Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Créez une pile pour cet exemple, comme suit. Vous pouvez aussi utiliser une pile Windows existante. Il vous suffit de mettre à jour les livres de recettes, comme décrit plus tard.

Création d'une pile

1. Ouvrez la console AWS OpsWorks Stacks et choisissez Add Stack (Ajouter une pile). Spécifiez les paramètres suivants, acceptez les valeurs par défaut pour les autres paramètres et choisissez Add Stack (Ajouter une pile).

- Nom — InstallIIS
- Région — Ouest des États-Unis (Oregon)

  Cet exemple fonctionne dans n'importe quelle région, mais nous vous recommandons d'utiliser US West (Oregon) pour les didacticiels.
- Système d'exploitation par défaut : Microsoft Windows Server 2012 R2

2. Choisissez Add a layer (Ajouter une couche) et ajoutez une couche personnalisée à la pile avec les paramètres suivants.

   - Nom — IIS
   - Nom court — iis

3. Ajoutez une instance 24/7 avec les paramètres par défaut de la couche IIS et démarrez-la.

Vous pouvez maintenant installer le livre de recettes et exécuter la recette.

Pour installer le livre de recettes et exécuter la recette

1. Modifiez la pile pour activer les livres personnalisés et spécifiez les paramètres suivants.

   - Type de référentiel — S3 Archive
   - URL du référentiel : URL de l'archive du livre de recettes que vous avez enregistrée précédemment.

   Acceptez les valeurs par défaut pour les autres paramètres, puis choisissez Save (Enregistrer) pour mettre à jour la configuration de la pile.

2. Exécutez la commande de pile Update Custom Cookbooks (Mettre à jour les livres de recettes personnalisées), qui installe la dernière version de vos livres de recettes personnalisées sur les instances en ligne de la pile. Si une version antérieure de vos livres de recettes est présente, cette commande la remplace.
3. Exécutez la recette à l'aide de la commande de pile Execute Recipes (Exécuter les recettes) après avoir défini Recipes to execute (Recettes à exécuter) sur **`install-iis::default`**. Cette commande lance une exécution de Chef, qui exécute les recettes spécifiées.

> **Note**
>
> Cet exemple utilise Execute Recipes pour des raisons pratiques, mais AWS OpsWorks Stacks exécute généralement vos recettes automatiquement en les affectant à l'événement du cycle de vie approprié. Vous pouvez exécuter ces recettes en déclenchant manuellement l'événement. Vous pouvez utiliser une commande de pile pour déclencher des événements Setup et Configure et une commande de déploiement pour déclencher des événements Deploy et Undeploy.

4. Pour vérifier l'installation, utilisez RDP afin de vous connecter à l'instance et ouvrez l'Explorateur Windows. Le système de fichiers doit maintenant comporter un répertoire `C:\inetpub`. Si vous vérifiez la liste des services de l'application du panneau de configuration des outils administratifs, IIS doit être vers le bas. Cependant, il sera nommé World Wide Web Publishing Service, et non IIS.

Installation d'un package sur une instance Windows

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cet exemple suppose que vous avez déjà fait l'exemple Exécution d'une recette sur une instance Windows. Si tel n'est pas le cas, commencez par cet exemple. Il décrit plus particulièrement comment autoriser l'accès RDP à vos instances.

Si votre logiciel est fourni dans un package d'installation, par exemple un fichier MSI, vous devez télécharger le fichier dans l'instance, puis l'exécuter. Cet exemple montre comment implémenter un livre de recettes pour installer un package MSI, l'exécution de Python, y compris la façon de

définir des variables d'environnement associées. Pour plus d'informations sur l'installation des fonctionnalités de Windows telles qu'IIS, consultez Installation d'une fonctionnalité de Windows : IIS.

Pour configurer le livre de recettes

1. Créez un répertoire nommé `installpython` et accédez à celui-ci.
2. Ajoutez un fichier `metadata.rb` à `installpython` avec le contenu suivant.

```
name "installpython"
version "0.1.0"
```

3. Ajoutez les répertoires `recipes` et `files` à `installpython`, et ajoutez un répertoire `default` aux fichiers.
4. Téléchargez un package Python à partir de Versions Python pour Windows et copiez-le dans le répertoire `files\default` du livre de recettes. Cet exemple installe la version Windows x86-de Python 3.5.0a3, qui utilise un programme d'installation nommé `python-3.4.3.amd64.msi`.
5. Ajoutez un fichier nommé `default.rb` au répertoire `recipes` avec le code suivant de la recette.

```
directory 'C:\tmp' do
  rights :full_control, 'Everyone'
  recursive true
  action :create
end

cookbook_file 'C:\tmp\python-3.4.3.amd64.msi' do
  source "python-3.4.3.amd64.msi"
  rights :full_control, 'Everyone'
  action :create
end

windows_package 'python' do
  source 'C:\tmp\python-3.4.3.amd64.msi'
  action :install
end

env "PATH" do
  value 'c:\python34'
  delim ";"
```

```
  action :modify
end
```

La recette exécute les tâches suivantes :

1. Utilisez une ressource directory pour créer un répertoire `C:\tmp`.

   Pour plus d'informations sur cette ressource, consultez Exemple 3 : Création de répertoires.

2. Utilisez une ressource cookbook_file pour copier le programme d'installation du répertoire `files\default` du livre de recettes vers `C:\tmp`.

   Pour plus d'informations sur cette ressource, consultez Installation d'un fichier à partir d'un livre de recettes.

3. Utilise une ressource windows_package pour exécuter le programme d'installation MSI, qui installe Python sur `c:\python34`.

   Le programme d'installation crée les répertoires nécessaires et installe les fichiers, mais il ne modifie pas la variable d'environnement `PATH` du système.

4. Utilise une ressource env pour ajouter `c:\python34` au chemin d'accès système.

   La ressource env vous permet de définir des variables d'environnement. Dans ce cas, la recette vous permet d'exécuter facilement des scripts Python à partir de la ligne de commande en ajoutant `c:\python34` au chemin d'accès.

   - Le nom de la ressource spécifie le nom de la variable d'environnement, `PATH` pour cet exemple.
   - L'attribut `value` spécifie la valeur de la variable, `c:\\python34` pour cet exemple (vous avez besoin d'échapper le caractère \).
   - L'action `:modify` ajoute la valeur spécifiée à la valeur actuelle de la variable.
   - L'attribut `delim` spécifie un délimiteur qui sépare la nouvelle valeur de la valeur existante, à savoir `;` pour cet exemple.

6. Créez une archive `.zip` du fichier `installpython`, chargez l'archive dans un compartiment S3 et rendez-la publique. Enregistrez l'URL de l'archive pour une utilisation ultérieure. Pour de plus amples informations, veuillez consulter Référentiels de livres de recettes.

   Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3

Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Créez une pile pour cet exemple, comme suit. Vous pouvez aussi utiliser une pile Windows existante. Il vous suffit de mettre à jour les livres de recettes, comme décrit plus tard.

Création d'une pile

1. Ouvrez la console AWS OpsWorks Stacks et choisissez Add Stack (Ajouter une pile). Spécifiez les paramètres suivants, acceptez les valeurs par défaut pour les autres paramètres et choisissez Add Stack (Ajouter une pile).

   - Nom — InstallPython
   - Région — Ouest des États-Unis (Oregon)

     Cet exemple fonctionne dans n'importe quelle région, mais nous vous recommandons d'utiliser US West (Oregon) pour les didacticiels.
   - Système d'exploitation par défaut : Microsoft Windows Server 2012 R2
2. Choisissez Add a layer (Ajouter une couche) et ajoutez une couche personnalisée à la pile avec les paramètres suivants.

   - Nom — Python
   - Nom court — python
3. Ajoutez une instance 24/7 avec les paramètres par défaut de la couche Python et démarrez-la.

Une fois que l'instance est en ligne, vous pouvez installer le livre de recettes et exécuter la recette

Pour installer le livre de recettes et exécuter la recette

1. Modifiez la pile pour activer les livres personnalisés et spécifiez les paramètres suivants.

   - Type de référentiel : archive S3.
   - URL du référentiel : URL de l'archive du livre de recettes que vous avez enregistrée précédemment.

   Acceptez les valeurs par défaut pour les autres paramètres, puis choisissez Save (Enregistrer) pour mettre à jour la configuration de la pile.

2. Exécutez la commande de pile Update Custom Cookbooks (Mettre à jour les livres de recettes personnalisées), qui installe la dernière version de vos livres de recettes personnalisées sur les instances en ligne de la pile. Si une version antérieure de votre livre de recettes est présente, cette commande la remplace.
3. Exécutez la recette à l'aide de la commande de pile Execute Recipes (Exécuter les recettes) après avoir défini Recipes to execute (Recettes à exécuter) sur **`installpython::default`**. Cette commande lance une exécution de Chef, avec une liste d'exécution composée de `installpython::default`.

   > **Note**
   >
   > Cet exemple utilise Execute Recipes pour des raisons pratiques, mais AWS OpsWorks Stacks exécute généralement vos recettes automatiquement en les affectant à l'événement du cycle de vie approprié. Vous pouvez exécuter ces recettes en déclenchant manuellement l'événement. Vous pouvez utiliser une commande de pile pour déclencher des événements Setup et Configure et une commande de déploiement pour déclencher des événements Deploy et Undeploy.

4. Pour vérifier l'installation, utilisez RDP afin de vous connecter à l'instance et ouvrez l'Explorateur Windows.

   - Le système de fichiers doit maintenant comporter un répertoire `C:\Python34`.
   - Si vous exécutez `path` à partir de la ligne de commande, il doit ressembler à : `PATH=c:\python34;C:\Windows\system32;...`
   - Si vous exécutez `python --version` à partir de la ligne de commande, il doit retourner `Python 3.4.3`.

Remplacement des attributs intégrés

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette rubrique s'applique uniquement aux piles Linux. Vous ne pouvez pas remplacer des attributs intégrés sur des piles Windows.

AWS OpsWorks Stacks installe un ensemble de livres de recettes intégrés sur chaque instance. La plupart des livres des recettes intégrés prennent en charge les couches intégrées, et leurs fichiers d'attributs définissent différents paramètres système et des applications par défaut, par exemple les paramètres de configuration du serveur Apache. En mettant ces paramètres dans des fichiers d'attributs, vous pouvez personnaliser de nombreux paramètres de configuration en remplaçant l'attribut intégré correspondant de l'une des façons suivantes :

- Définissez l'attribut dans le JSON personnalisé.

  Cette approche a l'avantage d'être simple et flexible. Cependant, vous devez entrer un JSON personnalisé manuellement, il n'y a donc aucune possibilité solide pour gérer les définitions d'attribut.
- Implémentez un livre de recettes personnalisé et définissez l'attribut dans un fichier d'attribut `customize.rb`.

  Cette approche est moins flexible que l'utilisation d'un JSON personnalisé, mais elle est plus solide car vous pouvez mettre vos livres de recettes personnalisés sous le contrôle de code source.

Cette rubrique décrit comment utiliser un fichier d'attributs de livre de recettes personnalisé pour remplacer les attributs intégrés, à l'aide du serveur Apache à titre d'exemple. Pour plus d'informations sur le remplacement des attributs avec un JSON personnalisé, consultez Utilisation du JSON personnalisé. Pour obtenir des informations générales sur le remplacement des attributs, consultez Remplacement des attributs.

> **Note**
>
> Le remplacement des attributs est le meilleur moyen de personnaliser les paramètres de configuration, mais les paramètres ne sont pas toujours représentés par des attributs. Dans ce cas, vous pouvez souvent personnaliser le fichier de configuration en remplaçant le modèle que les recettes intégrées utilisent pour créer le fichier de configuration. Pour obtenir un exemple, consultez Remplacement des modèles intégrés.

Les attributs intégrés représentent généralement des valeurs dans les fichiers de modèle que les recettes Setup utilisent pour créer des fichiers de configuration. Par exemple, l'une des recettes Setup `apache2`, `default.rb`, utilise le modèle `apache2.conf.erb` pour créer le fichier de configuration principal du serveur Apache, `httpd.conf` (Amazon Linux) ou `apache2.conf` (Ubuntu). Voici un extrait du modèle de fichier :

```
...
#
# 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)
##

...
```

Le paramètre `KeepAliveTimeout` de cet exemple est la valeur de l'attribut `[:apache][:keepalivetimeout]`. La valeur par défaut de cet attribut est définie dans le livre de recettes `apache2`et son fichier d'attributs `apache.rb`, comme illustré dans l'extrait suivant :

```
...
# 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**
>
> Pour plus d'informations sur les attributs intégrés couramment utilisés, consultez Attributs des livres de recettes intégrés.

Pour prendre en charge le remplacement des attributs intégrés, tous les livres de recettes intégrés comportent un fichier d'attributs `customize.rb` qui est intégré à tous les modules via une directive `include_attribute`. Les fichiers `customize.rb` des livres de recettes intégrés ne contiennent aucune définition d'attribut et n'ont aucun effet sur les attributs intégrés. Pour remplacer les attributs intégrés, vous créez un livre de recettes personnalisé avec le même nom que le livre de recettes intégré et vous placez vos définitions d'attributs personnalisés dans un fichier d'attributs également nommé `customize.rb`. Ce fichier est prioritaire sur la version intégrée et il est inclus dans les modules associés. Si vous définissez les attributs intégrés dans votre fichier `customize.rb`, ils remplacent les attributs intégrés correspondants.

Cet exemple montre comment remplacer l'attribut intégré `[:apache][:keepalivetimeout]` pour définir sa valeur sur 5 au lieu de 3. Vous pouvez utiliser une approche similaire pour n'importe quel attribut intégré. Cependant, soyez prudent avec les attributs que vous remplacez. Par exemple, le remplacement des attributs de l'espace de noms `opsworks` peut entraîner des problèmes pour certaines recettes intégrées.

> **Important**
>
> Ne remplacez pas les attributs intégrés en modifiant une copie du fichier d'attributs intégrés lui-même. Par exemple, vous pouvez placer une copie du fichier `apache.rb` dans le dossier `apache2/attributes` de votre livre de recettes personnalisées, et modifier certains de ses paramètres. Toutefois, ce fichier est prioritaire sur la version intégrée et les recettes intégrées utiliseront maintenant votre version du fichier `apache.rb`. Si AWS OpsWorks Stacks modifie ultérieurement le `apache.rb` fichier intégré, les recettes n'obtiendront pas les nouvelles valeurs à moins que vous ne mettiez à jour manuellement votre version. En utilisant`customize.rb`, vous remplacez uniquement les attributs spécifiés ; les recettes intégrées continuent à obtenir automatiquement des up-to-date valeurs pour chaque attribut que vous n'avez pas remplacé.

Pour commencer, créez un livre de recettes personnalisé.

Pour créer le livre de recettes

1. Dans votre répertoire `opsworks_cookbooks`, créez un répertoire de livres de recettes nommé `apache2` et accédez à celui-ci.

   Pour que vous puissiez remplacer les attributs intégrés, le livre de recettes personnalisé doit avoir le même nom que le livre de recettes intégré, `apache2` pour cet exemple.

2. Dans le répertoire `apache2`, créez un répertoire `attributes`.

3. Ajoutez un fichier nommé `customize.rb` au répertoire `attributes` et utilisez-le pour définir le livre de recettes intégré qui vous souhaitez remplacer. Pour cet exemple, le fichier doit contenir les éléments suivants :

   ```
   normal[:apache][:keepalivetimeout] = 5
   ```

   > **Important**
   >
   > Pour que vous puissiez remplacer un attribut intégré, un attribut personnalisé doit avoir le type `normal` ou porter exactement le même nom de nœud que l'attribut intégré correspondant. Le type `normal` garantit que l'attribut personnalisé est prioritaire sur les attributs intégrés, qui sont tous de type `default`. Pour de plus amples informations, veuillez consulter Priorité des attributs.

4. Créez une `.zip` archive `opsworks_cookbooks` nommée `opsworks_cookbooks.zip` et chargez-la dans un compartiment Amazon Simple Storage Service (Amazon S3). Pour des raisons de simplicité, rendez le fichier public. Enregistrez l'URL pour une utilisation ultérieure. Vous pouvez également stocker vos livres de recettes dans une archive privée Amazon S3 ou dans d'autres types de référentiels. Pour de plus amples informations, veuillez consulter Référentiels de livres de recettes.

   Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Pour utiliser l'attribut personnalisé, créez une pile et installez le livre de recettes.

Pour utiliser l'attribut personnalisé

1. Ouvrez la console AWS OpsWorks, puis choisissez Add Stack (Ajouter une pile).
2. Spécifiez les paramètres standard suivants.

   - Nom — ApacheConfig
   - Région — Ouest des États-Unis (Oregon)

     Vous pouvez placer votre stack dans n'importe quelle région, mais nous vous recommandons l'ouest des États-Unis (Oregon) pour les tutoriels.
   - Clé SSH par défaut : une paire de EC2 clés

     Si vous devez créer une paire de EC2 clés, consultez Amazon EC2 Key Pairs. Notez que la paire de clés doit appartenir à la même région AWS que la pile.

   Choisissez Advanced>> (Avancé>>), définissez Use custom Chef cookbooks (Utiliser les livres de recettes Chef personnalisés) sur Yes (Oui), puis spécifiez les paramètres suivants.

   - Type de référentiel — Http Archive
   - URL du référentiel : URL de l'archive du livre de recettes que vous avez enregistrée précédemment

   Acceptez les valeurs par défaut pour les autres paramètres, puis choisissez Add Stack (Ajouter une pile) pour créer la pile.

   > **Note**
   >
   > Cet exemple utilise le système d'exploitation par défaut, Amazon Linux. Vous pouvez utiliser Ubuntu si vous le souhaitez. La seule différence est que sur les systèmes Ubuntu, la recette Setup intégrée produit un fichier de configuration avec les mêmes paramètres nommé `apache2.conf` et le place dans le répertoire `/etc/apache2`.
3. Choisissez Ajouter une couche, puis ajoutez une couche Java App Server avec les paramètres par défaut à la pile.
4. Ajoutez une instance 24/7 avec les paramètres par défaut dans la couche, puis lancez l'instance.

   Une instance t2.micro suffit pour cet exemple.

5. Une fois que l'instance est en ligne, connectez-la avec SSH. Le fichier `httpd.conf` se trouve dans le répertoire `/etc/httpd/conf`. Si vous examinez le fichier, vous devez voir votre paramètre personnalisé `KeepAliveTimeout`. Les autres paramètres auront les valeurs par défaut du fichier intégré `apache.rb`. La partie appropriée de `httpd.conf` doit être similaire à ce qui suit :

```
...
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 5
...
```

Remplacement des modèles intégrés

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette rubrique s'applique uniquement aux piles Linux. Vous ne pouvez pas remplacer des modèles intégrés sur des piles Windows.

Les recettes intégrées de AWS OpsWorks Stacks utilisent des modèles pour créer des fichiers sur des instances, principalement des fichiers de configuration pour des serveurs, tels qu'Apache. Par exemple, les recettes `apache2` utilisent le modèle `apache2.conf.erb` pour créer le fichier de configuration principal du serveur Apache, `httpd.conf` (Amazon Linux) ou `apache2.conf` (Ubuntu).

La plupart des paramètres de configuration de ces modèles sont représentés par des attributs, le meilleur moyen de personnaliser un fichier de configuration est donc de remplacer les attributs intégrés appropriés. Pour obtenir un exemple, consultez Remplacement des attributs intégrés. Toutefois, si les paramètres que vous souhaitez personnaliser ne sont pas représentés par des attributs intégrées, ou s'ils ne sont pas du tout dans le modèle, vous devez remplacer le modèle lui-même. Cette rubrique décrit comment remplacer un modèle intégré pour spécifier un paramètre de configuration personnalisé Apache.

Vous pouvez fournir des réponses d'erreur personnalisées à Apache en ajoutant les paramètres `ErrorDocument` au fichier `httpd.conf`. `apache2.conf.erb` contient uniquement des exemples mis en commentaire, comme indiqué ci-après :

```
...
#
# 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
...
```

Etant donné que ces paramètres sont des commentaires codés en dur, vous ne pouvez pas spécifier de valeurs personnalisées en remplaçant les attributs ; vous devez remplacer le modèle lui-même. Cependant, contrairement à ce qui se passe avec les attributs, il n'est pas possible de remplacer certaines parties d'un modèle de fichier. Vous devez créer un livre de recettes personnalisé portant le même nom que la version intégrée, copier le modèle de fichier dans le même sous-répertoire et modifier le fichier en fonction des besoins. Cette rubrique montre comment remplacer `apache2.conf.erb` pour fournir une réponse personnalisée à une erreur 500. Pour plus d'informations générales sur le remplacement des modèles, consultez Utilisation de modèles personnalisés.

> **Important**
>
> Lorsque vous remplacez un modèle intégré, les recettes intégrées utilisent votre version personnalisée du modèle au lieu de la version intégrée. Si AWS OpsWorks Stacks met à jour le modèle intégré, le modèle personnalisé devient désynchronisé et risque de

ne pas fonctionner correctement. AWS OpsWorks Stacks n'apporte pas souvent de telles modifications, et lorsqu'un modèle change, AWS OpsWorks Stacks répertorie les modifications et vous donne la possibilité de passer à une nouvelle version. Nous vous recommandons de surveiller le référentiel AWS OpsWorks Stacks pour détecter des modifications et de mettre à jour manuellement votre modèle personnalisé en fonction des besoins. Notez que le référentiel a une branche distincte pour chaque version de Chef prise en charge, alors veillez à utiliser la branche appropriée.

Pour commencer, créez un livre de recettes personnalisé.

Pour créer le livre de recettes

1. Dans le répertoire `opsworks_cookbooks`, créez un répertoire de livres de recettes nommé `apache2` et accédez à celui-ci. Pour que vous puissiez remplacer les modèles intégrés, le livre de recettes personnalisé doit avoir le même nom que le livre de recettes intégré, `apache2` pour cet exemple.

   > **Note**
   >
   > Si vous avez déjà découvert la procédure Remplacement des attributs intégrés, vous pouvez utiliser le même livre de recettes `apache2` pour cet exemple et ignorer l'étape 2.

2. Créez un fichier `metadata.rb` avec le contenu suivant, puis enregistrez-le dans le répertoire `apache2`.

   ```
   name "apache2"
   version "0.1.0"
   ```

3. Dans le répertoire `apache2`, créez un répertoire `templates/default`.

   > **Note**
   >
   > Le `templates/default` répertoire fonctionne pour les instances Amazon Linux, qui utilisent le `apache2.conf.erb` modèle par défaut. Les instances Ubuntu 14.04 utilisent un modèle `apache2.conf.erb` propre au système d'exploitation et se trouvant dans le répertoire `templates/ubuntu-14.04`. Si vous souhaitez que la personnalisation

s'applique également aux instances Ubuntu 14.04, vous devez également remplacer ce modèle.

4. Copiez le modèle `apache2.conf.erb` intégré dans votre répertoire `templates/default`. Ouvrez le fichier du modèle, supprimez le commentaire de la ligne `ErrorDocument 500` et entrez un message d'erreur personnalisé, comme suit :

```
...
ErrorDocument 500 "A custom error message."
#ErrorDocument 404 /missing.html
...
```

5. Créez une `.zip` archive `opsworks_cookbooks` nomméeopsworks_cookbooks.zip, puis chargez le fichier dans un compartiment Amazon Simple Storage Service (Amazon S3). Pour simplifier les choses, rendez l'archive publique. Enregistrez l'URL de l'archive pour une utilisation ultérieure. Vous pouvez également stocker vos livres de recettes dans une archive privée Amazon S3 ou dans d'autres types de référentiels. Pour de plus amples informations, veuillez consulter Référentiels de livres de recettes.

   Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

> **Note**
>
> Pour plus de simplicité, cet exemple ajoute un message d'erreur codé en dur au modèle. Pour le changer, vous devez modifier le modèle et réinstaller le livre de recettes. Afin de bénéficier d'une plus grande souplesse, vous pouvez définir un attribut personnalisé par défaut pour la chaîne d'erreur dans le fichier d'attribut `customize.rb` du livre de recettes personnalisées et attribuer la valeur de cet attribut à `ErrorDocument 500`. Par exemple, si vous nommez l'attribut `[:apache][:custom][:error500]`, la ligne correspondante dans `apache2.conf.erb` ressemble à ce qui suit :
>
> ```
> ...
> ErrorDocument 500 <%= node[:apache][:custom][:error500] %>
> ```

```
#ErrorDocument 404 /missing.html
...
```

Vous pouvez ensuite modifier le message d'erreur personnalisé à tout moment en remplaçant `[:apache][:custom][:error500]`. Si vous utilisez le JSON personnalisé pour remplacer l'attribut, vous n'avez même pas besoin de toucher le livre de recettes.

Pour utiliser le modèle personnalisé, créez une pile et installez le livre de recettes.

Pour utiliser le modèle personnalisé

1. Ouvrez la console AWS OpsWorks, puis choisissez Add Stack (Ajouter une pile).
2. Spécifiez les paramètres standard suivants :

   - Nom — ApacheTemplate
   - Région — Ouest des États-Unis (Oregon)
   - Clé SSH par défaut : une paire de clés Amazon Elastic Compute Cloud (Amazon EC2)

     Si vous devez créer une paire de EC2 clés Amazon, consultez Amazon EC2 Key Pairs. Notez que la paire de clés doit appartenir à la même région AWS que l'instance.

   Choisissez Advanced>> (Avancé>>), puis Use custom Chef cookbooks (Utiliser les livres de recettes Chef personnalisés) pour spécifier les paramètres suivants :

   - Type de référentiel — Http Archive
   - URL du référentiel : URL de l'archive du livre de recettes que vous avez enregistrée précédemment

   Acceptez les valeurs par défaut pour les autres paramètres, puis choisissez Add Stack (Ajouter une pile) pour créer la pile.
3. Choisissez Ajouter une couche, puis ajoutez une couche Java App Server à la pile avec les paramètres par défaut.
4. Ajoutez une instance 24/7 avec les paramètres par défaut dans la couche, puis lancez l'instance.

   Une instance t2.micro suffit pour cet exemple.

5. Une fois que l'instance est en ligne, connectez-la avec SSH. Le fichier `httpd.conf` se trouve dans le répertoire `/etc/httpd/conf`. Le fichier doit contenir le paramètre personnalisé `ErrorDocument`, qui doit ressembler à ce qui suit :

```
...
# 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
...
```

Equilibrage de charge d'une couche

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks propose deux options d'équilibrage de charge, Elastic Load Balancing et Elastic Load Balancing HAProxy, qui sont généralement utilisées pour équilibrer la charge entre les instances d'une couche de serveur d'applications. Cette rubrique décrit les avantages et les limites de chacune pour vous aider à déterminer quelle option choisir lors de l'ajout d'un équilibrage de charge à une couche. Dans certains cas, la meilleure approche consiste à utiliser les deux.

Résiliation SSL

La HAProxy couche intégrée ne gère pas la terminaison SSL ; vous devez résilier le SSL sur les serveurs. L'avantage de cette approche est que le trafic est chiffré jusqu'à ce qu'il atteigne les serveurs. Toutefois, les serveurs doivent gérer le déchiffrement, ce qui augmente la charge du serveur. En outre, vous devez mettre vos certificats SSL sur les serveurs d'applications, qui sont plus accessibles aux utilisateurs.

Avec Elastic Load Balancing, vous pouvez mettre fin au protocole SSL au niveau de l'équilibreur de charge. Cela réduit la charge sur vos serveurs d'applications, mais le trafic entre l'équilibreur de charge et le serveur n'est pas chiffré. Elastic Load Balancing vous permet également de mettre fin au protocole SSL sur le serveur, mais sa configuration est un peu compliquée.

Mise à l'échelle

Si le trafic entrant dépasse la capacité d'un équilibreur de HAProxy charge, vous devez augmenter sa capacité manuellement.

Elastic Load Balancing s'adapte automatiquement pour gérer le trafic entrant. Pour vous assurer qu'un équilibreur de charge Elastic Load Balancing dispose d'une capacité suffisante pour gérer la charge attendue lors de sa première mise en ligne, vous pouvez le préchauffer.

Échec de l'équilibreur de charge

Si l'instance hébergeant votre HAProxy serveur tombe en panne, l'ensemble de votre site peut être mis hors ligne jusqu'à ce que vous puissiez redémarrer l'instance.

Elastic Load Balancing est plus résistant aux défaillances que HAProxy. Par exemple, il fournit des nœuds d'équilibrage de charge dans chaque zone de disponibilité contenant EC2 des instances enregistrées. Si le service d'une zone est interrompu, les autres nœuds continuent à gérer le trafic entrant. Pour plus d'informations, consultez Elastic Load Balancing Concepts.

Délai d'inactivité

Les deux équilibreurs de charge mettent une connexion hors service si un serveur est inactif pendant plus d'une valeur de délai d'inactivité spécifiée.

- HAProxy — La valeur du délai d'inactivité n'a pas de limite supérieure.
- Elastic Load Balancing — La valeur du délai d'inactivité par défaut est de 60 secondes, avec un maximum de 3 600 secondes (60 minutes).

La limite de temps d'inactivité d'Elastic Load Balancing est suffisante dans la plupart des cas. Nous vous recommandons de l'utiliser HAProxy si vous avez besoin d'un délai d'inactivité plus long. Par exemple :

- Une connexion HTTP de longue durée qui est utilisée pour les notifications push.
- Une interface d'administration que vous utilisez pour exécuter des tâches susceptibles de durer un maximum de 60 minutes.

Mappage basé sur les URL

Vous pouvez avoir besoin qu'un équilibreur de charge transmette une demande entrante à un serveur particulier reposant sur l'URL de la demande. Par exemple, supposons que vous ayez un groupe de dix serveurs d'applications qui prend en charge une application de commerce en ligne. Huit des serveurs gèrent le catalogue et deux gèrent les paiements. Vous voulez transférer toutes les requêtes HTTP liées aux paiements aux serveurs de paiement, en fonction de l'URL de la demande. Dans ce cas, vous dirigerez tout ce URLs qui inclut le « paiement » ou le « paiement » vers l'un des serveurs de paiement.

Avec HAProxy, vous pouvez utiliser le mappage basé sur les URL pour diriger le contenu d' URLs une chaîne spécifiée vers des serveurs particuliers. Pour utiliser le mappage basé sur des URL avec AWS OpsWorks Stacks, vous devez créer un fichier de HAProxy configuration personnalisé en remplaçant le `haproxy-default.erb` modèle dans le livre de recettes intégré. `haproxy` Pour plus d'informations, consultez le manuel HAProxy de configuration etUtilisation de modèles personnalisés. Vous ne pouvez pas utiliser le mappage basé sur les URL pour les requêtes HTTPS. Une requête HTTPS est cryptée et n' HAProxya donc aucun moyen d'examiner l'URL de la demande.

La prise en charge du mappage d'URL par Elastic Load Balancing est limitée. Pour de plus amples informations, veuillez consulter Écouteurs de votre Classic Load Balancer.

Recommandation : Nous vous recommandons d'utiliser Elastic Load Balancing pour l'équilibrage de charge, sauf si vous avez des exigences qui ne peuvent être traitées que par HAProxy. Dans ce cas, la meilleure approche consiste peut-être à combiner les deux en utilisant Elastic Load Balancing comme équilibreur de charge frontal qui distribue le trafic entrant vers un ensemble de HAProxy serveurs. Pour cela :

- Configurez une HAProxy instance dans chacune des zones de disponibilité de votre stack pour distribuer les demandes aux serveurs d'applications de la zone.
- Assignez les HAProxy instances à un équilibreur de charge Elastic Load Balancing, qui distribue ensuite les demandes entrantes aux équilibreurs de HAProxy charge.

Cette approche vous permet d'utiliser HAProxy le mappage basé sur les URL pour distribuer différents types de demandes aux serveurs d'applications appropriés. Toutefois, si l'un des HAProxy serveurs est hors ligne, le site continuera de fonctionner car l'équilibreur de charge Elastic Load Balancing distribue automatiquement le trafic entrant aux HAProxy serveurs sains. Notez que vous

devez utiliser Elastic Load Balancing comme équilibreur de charge frontal ; un HAProxy serveur ne peut pas distribuer de demandes à d'autres HAProxy serveurs.

Migration de Chef Server vers Stacks AWS OpsWorks

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks étant basé sur Chef, la migration de Chef Server vers AWS OpsWorks Stacks est relativement simple. Cette rubrique présente les directives relatives à la modification du code Chef Server pour qu'il fonctionne avec AWS OpsWorks Stacks.

> **Note**
>
> Il est déconseillé de migrer vers les piles à l'aide des versions Chef antérieures à 11.10, car elles sont basées sur chef-solo et ne prennent pas en charge la recherche ou les conteneurs de données.

Rubriques

- Mappage des rôles et des couches
- Utilisation des conteneurs de données
- Utilisation de Chef Search
- Gestion des recettes et des livres de recettes
- Utilisation des environnements Chef

Mappage des rôles et des couches

Chef Server utilise des rôles pour représenter et gérer les instances ayant les mêmes objectif et configuration, telles qu'un ensemble d'instances qui hébergent chacune un serveur d'applications Java. Une couche AWS OpsWorks Stacks obéit pour l'essentiel au même objectif qu'un rôle Chef.

Une couche est un modèle permettant de créer un ensemble d'instances Amazon Elastic Compute Cloud (Amazon EC2) avec la même configuration, les mêmes packages installés, la même procédure de déploiement d'applications, etc.

AWS OpsWorks Stacks inclut un ensemble de couches intégrées pour plusieurs types de serveurs d'applications, un équilibreur de HAProxy charge, un maître de base de données MySQL et un maître de surveillance Ganglia. Par exemple, la couche Java App Server intégrée est un modèle pour créer des instances hébergeant un serveur Tomcat.

Pour migrer vers AWS OpsWorks Stacks, vous devez associer chaque rôle à une couche fournissant des fonctionnalités équivalentes. Pour certains rôles, il se peut que vous puissiez simplement utiliser l'une des couches intégrées. D'autres rôles peuvent nécessiter différents degrés de personnalisation. Commencez par examiner les fonctionnalités des couches intégrées, y compris les recettes associées à chacune d'elles, pour voir si l'une offre au moins certaines fonctionnalités de votre rôle. Pour plus d'informations sur les couches intégrées, consultez Couches et AWS OpsWorks Référence de la couche Stacks. Pour examiner les recettes intégrées, consultez le GitHub référentiel public AWS OpsWorks Stacks.

La façon dont vous procédez dépend étroitement de la manière dont vous pouvez associer une couche à chaque rôle, comme suit.

Une couche intégrée prend en charge toutes les fonctionnalités du rôle

> Vous pouvez utiliser la couche intégrée directement, avec des personnalisations mineures, si nécessaire. Par exemple, si un rôle prend en charge un serveur Tomcat, les recettes de la couche Java App Server peuvent déjà gérer toutes les tâches du rôle, éventuellement avec une légère personnalisation. Par exemple, vous pouvez faire en sorte que les recettes intégrées de la couche utilisent les paramètres de configuration Tomcat ou Apache en remplaçant les attributs ou les modèles appropriés.

Une couche intégrée prend en charge certaines, mais pas toutes, fonctionnalités du rôle

> Vous pouvez utiliser une couche intégrée en étendant la couche. Cette extension implique généralement d'implémenter des recettes personnalisées pour prendre en charge les fonctionnalités manquantes et d'affecter les recettes aux événements du cycle de vie de la couche. Par exemple, supposons que votre rôle installe un serveur Redis sur les mêmes instances qui hébergent un serveur Tomcat. Vous pouvez étendre la couche Java App Server pour qu'elle corresponde aux fonctionnalités du rôle en implémentant une recette personnalisée pour installer Redis sur les instances de la couche et en attribuant la recette à l'événement de configuration de la couche.

Aucune couche intégrée ne prend en charge de façon adéquate la fonctionnalité du rôle

Mettez en place une couche personnalisée. Par exemple, supposons que votre rôle prenne en charge un serveur de base de données MongoDB, qui n'est pris en charge par aucune des couches intégrées. Vous pouvez fournir cette prise en charge en implémentant des recettes pour installer les packages requis, configurer le serveur, etc., et attribuer les recettes aux événements du cycle de vie d'une couche personnalisée. Généralement, vous pouvez utiliser au moins certaines recettes du rôle à cet effet. Pour plus d'informations sur l'implémentation d'une couche personnalisée, consultez Création d'une couche serveur Tomcat personnalisée.

Utilisation des conteneurs de données

Chef Server vous permet de transmettre à vos recettes des données définies par l'utilisateur à l'aide de conteneurs de données.

- Vous stockez les données avec vos livres de recettes et Chef les installe sur chaque instance.
- Vous pouvez utiliser les conteneurs de données chiffrées pour les données sensibles telles que les mots de passe.

AWS OpsWorks Stacks prend en charge les sacs de données ; les recettes peuvent récupérer les données en utilisant exactement le même code qu'avec Chef Server. Cependant, la prise en charge présente les différences et limites suivantes :

- Les conteneurs de données ne sont pris en charge que sur les piles Linux Chef 11 .10 et versions ultérieures.

  Les piles Windows et les piles Linux exécutant des versions antérieures de Chef ne prennent pas en charge les conteneurs de données.
- Vous ne stockez pas les conteneurs de données dans votre référentiel de livres de recettes.

  Au lieu de cela, vous utilisez un JSON personnalisé pour gérer les données de vos conteneurs de données.
- AWS OpsWorks Stacks ne prend pas en charge les sacs de données cryptés.

  Si vous devez stocker des données sensibles sous une forme chiffrée, telle que mots de passe ou certificats, nous vous recommandons de les stocker dans un compartiment S3 privé. Vous pouvez ensuite créer une recette personnalisée qui utilise le Kit SDK Amazon pour Ruby pour récupérer les données. Pour obtenir un exemple, consultez Utilisation du kit SDK pour Ruby.

Pour de plus amples informations, veuillez consulter Utilisation des conteneurs de données.

Utilisation de Chef Search

Chef Server stocke les informations de configuration de pile, telles que les adresses IP et les configurations de rôle, sur le serveur. Les recettes utilisent la recherche Chef pour récupérer ces données. AWS OpsWorks Stacks utilise une approche quelque peu différente. Par exemple, les piles Linux Chef 11.10 reposent sur le mode local client Chef, option qui exécute localement une version légère de Chef Server (souvent appelé Chef Zero) sur l'instance. Chef Zero prend en charge la recherche sur les données stockées dans l'objet nœud de l'instance.

Au lieu de stocker les données de pile sur un serveur distant, AWS OpsWorks Stacks ajoute un ensemble d'attributs de configuration et de déploiement de la pile à l'objet nœud de chaque instance pour chaque événement du cycle de vie. Ces attributs représentent un instantané de la configuration de la pile. Ils utilisent la même syntaxe que Chef Server et représentent la plupart des données dont les recettes ont besoin pour récupérer des données à partir du serveur.

Vous n'avez souvent pas besoin de modifier le code de recherche de vos recettes pour Stacks. AWS OpsWorks Comme Chef search fonctionne sur l'objet nœud, qui inclut la configuration de la pile et les attributs de déploiement, les requêtes de recherche dans AWS OpsWorks Stacks fonctionnent généralement exactement comme elles le font avec Chef Server.

La principale exception est due au fait que les attributs de configuration et de déploiement de la pile contiennent uniquement des données dont AWS OpsWorks Stacks a connaissance lorsqu'il installe les attributs sur l'instance. Si vous créez ou modifiez un attribut localement sur une instance particulière, ces modifications ne sont pas répercutées vers AWS OpsWorks Stacks et ne sont pas incorporées dans la configuration de la pile et les attributs de déploiement installés sur les autres instances. Vous pouvez utiliser la recherche pour récupérer la valeur d'attribut uniquement sur cette instance. Pour de plus amples informations, veuillez consulter Utilisation de Chef Search.

Pour des raisons de compatibilité avec Chef Server, AWS OpsWorks Stacks ajoute un ensemble d'`role`attributs à l'objet nœud, chacun contenant l'un des attributs de couche de la pile. Si votre recette utilise `roles` comme clé de recherche, vous n'avez pas besoin de modifier le code de recherche. La requête retourne automatiquement les données de la couche correspondante. Par exemple, les requêtes suivantes interrogent toutes deux les attributs de la couche `php-app`.

```
phpserver = search(:node, "layers:php-app").first
```

```
phpserver = search(:node, "roles:php-app").first
```

Gestion des recettes et des livres de recettes

AWS OpsWorks Stacks et Chef Server gèrent les livres de cuisine et les recettes de manière quelque peu différente. Avec Chef Server :

- Vous fournissez tous les livres de recettes, en les mettant en œuvre vous-même ou en utilisant les livres de recettes de la communauté.
- Vous stockez les livres de recettes sur le serveur.
- Vous exécutez les recettes manuellement ou selon une planification régulière.

Avec AWS OpsWorks Stacks :

- AWS OpsWorks Stacks fournit un ou plusieurs livres de recettes pour chacune des couches intégrées. Ces livres de recettes gèrent les tâches standards, telles que l'installation et la configuration des logiciels d'une couche intégrée, et le déploiement d'applications.

  Pour gérer des tâches qui ne sont pas exécutées par les livres de recettes intégrés, vous ajoutez des livres de recettes personnalisés à votre pile ou utilisez les livres de recettes de la communauté.
- Vous stockez les livres de recettes AWS OpsWorks Stacks dans un référentiel distant, tel qu'un compartiment S3 ou un dépôt Git.

  Pour de plus amples informations, veuillez consulter Stockage des livres de recettes.
- Vous pouvez exécuter des recettes manuellement, mais AWS OpsWorks Stacks exécute généralement des recettes pour vous en réponse à un ensemble d'événements du cycle de vie qui se produisent à des moments clés du cycle de vie d'une instance.

  Pour de plus amples informations, veuillez consulter Exécution des recettes.
- AWS OpsWorks Stacks prend uniquement en charge les stacks Berkshelf on Chef 11.10. Si vous utilisez Berkshelf pour gérer les dépendances de vos livres de recettes, vous ne pouvez pas utiliser les piles exécutant Chef 11.4 ou versions antérieures.

  Pour de plus amples informations, veuillez consulter Utilisation de Berkshelf.

Rubriques

- Stockage des livres de recettes
- Exécution des recettes

Stockage des livres de recettes

Avec Chef Server, vous stockez vos livres de recettes sur le serveur et les déployez depuis le serveur vers les instances. Avec AWS OpsWorks Stacks, vous stockez des livres de recettes dans un dépôt : une archive S3 ou HTTP ou un dépôt Git ou Subversion. Vous spécifiez les informations dont AWS OpsWorks Stacks a besoin pour télécharger le code du référentiel vers les instances d'une pile lorsque vous installez des livres de recettes.

Pour migrer à partir de Chef Server, vous devez placer vos livres de recettes dans l'un de ces référentiels. Pour plus d'informations sur la façon de structurer un référentiel de livres de recettes, consultez Référentiels de livres de recettes.

Exécution des recettes

Dans AWS OpsWorks Stacks, chaque couche est associée à un ensemble d'événements du cycle de vie (installation, configuration, déploiement, annulation du déploiement et arrêt) qui se produisent chacun à un moment clé du cycle de vie d'une instance. Pour exécuter une recette personnalisée, vous l'assignez généralement à l'événement approprié de la couche appropriée. Lorsque l'événement se produit, AWS OpsWorks Stacks exécute les recettes associées. Par exemple, l'événement Setup se produit après qu'une instance a fini de démarrer et, par conséquent, vous lui assignez généralement les recettes qui exécutent des tâches telles que l'installation et la configuration de packages, ou le démarrage de services.

Vous pouvez exécuter les recettes manuellement grâce à la commande de pileExecute Recipes. Cette commande est utile pour le développement et les tests, mais vous pouvez également l'utiliser pour exécuter des recettes qui ne sont pas mappées avec un événement du cycle de vie. Vous pouvez aussi utiliser la commande Execute Recipes pour déclencher manuellement les événements Setup et Configure.

Outre la console AWS OpsWorks Stacks, vous pouvez utiliser l'interface de ligne de commande AWS ou SDKsexécuter des recettes. Ces outils prennent en charge toutes les actions d'API AWS OpsWorks Stacks, mais sont plus simples à utiliser que l'API. Utilisez la commande create-deployment de l'interface de ligne de commande pour déclencher un événement de cycle de vie, qui exécute toutes les recettes associées. Vous pouvez également utiliser cette commande pour exécuter une ou plusieurs recettes sans déclencher un événement. Le code équivalent du kit de

développement logiciel dépend de la langue, mais il est généralement similaire à la commande de la ligne de commande.

Les exemples suivants décrivent deux façons d'utiliser la commande `create-deployment` de l'interface de ligne de commande pour automatiser le déploiement d'applications.

- Déployez votre application selon un calendrier régulier en ajoutant à votre pile une couche personnalisée avec une seule instance.

  Ajoutez à la couche une recette Setup personnalisée qui crée un travail `cron` sur l'instance pour exécuter la commande selon un calendrier spécifié. Pour obtenir un exemple d'utilisation d'une recette pour créer un travail `cron`, consultez Exécution de tâches cron sur les instances Linux.
- Ajoutez une tâche à votre pipeline d'intégration continue qui utilise la commande `create-deployment` de l'interface de ligne de commande pour déployer l'application.

Utilisation des environnements Chef

AWS OpsWorks Stacks ne prend pas en charge les environnements Chef ; revient `node.chef_environment` `_default` toujours.

AWS OpsWorks Référence de la couche Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Chaque instance déployée par AWS OpsWorks Stacks doit être membre d'au moins une couche, qui définit le rôle d'une instance dans la pile et contrôle les détails de l'installation et de la configuration de l'instance, de l'installation des packages, du déploiement des applications, etc. Pour plus d'informations sur l'utilisation des AWS OpsWorks Stacks pour créer et gérer des couches, consultezCouches.

La description de chaque couche inclut une liste des recettes intégrées que AWS OpsWorks Stacks exécute pour chacun des événements du cycle de vie de la couche. Ces recettes sont stockées sur

https://github.com/aws/opsworks-cookbooks. Notez que les listes incluent uniquement les recettes exécutées directement par AWS OpsWorks Stacks. Ces recettes exécutent parfois des recettes dépendantes, qui ne sont pas répertoriées. Pour voir la liste complète des recettes d'un événement particulier, y compris les recettes dépendantes et les recettes personnalisées, examinez la liste d'exécution dans le journal Chef de l'événement du cycle de vie.

Rubriques

- HAProxy Référence de couche
- HAProxy AWS OpsWorks Couche Stacks
- Référence de couche MySQL
- OpsWorks Couche MySQL
- Référence des couches serveur d'applications
- Couches serveur d'applications
- Référence de couche de cluster ECS
- Référence de couche personnalisée
- Autres couches de référence
- Autres couches

HAProxy Référence de couche

Important

Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Note

Cette couche est disponible uniquement pour les piles Linux.

Une HAProxy couche utilise HAProxyun équilibreur de TCP/HTTP charge fiable et performant pour fournir des services d'équilibrage de charge et de proxy à haute disponibilité pour les applications TCP et HTTP. Elle est particulièrement utile pour les sites web qui doivent analyser des charges très élevées, tout en exigeant un traitement permanent ou de couche 7.

HAProxy surveille le trafic et affiche les statistiques et l'état de santé des instances associées sur une page Web. Par défaut, l'URI est http ://*DNSName*/haproxy ? stats, où *DNSName* est le nom DNS de l' HAProxy instance.

Nom court : lb

Compatibilité : une HAProxy couche est compatible avec les couches suivantes : custom, db-master et memcached.

Ports ouverts : HAProxy autorise l'accès public aux ports 22 (SSH), 80 (HTTP) et 443 (HTTPS).

Autoassign Elastic IP addresses : Activé par défaut

Default EBS volume : Non

Groupe de sécurité par défaut : AWS-OpsWorks-LB-Server

Configuration : Pour configurer une HAProxy couche, vous devez spécifier les éléments suivants :

- URI de contrôle de santé (par défaut : http ://*DNSName*/).
- URI des statistiques (par défaut : http ://*DNSName*/haproxy ? statistiques).
- Mot de passe des statistiques (facultatif).
- Méthode de contrôle de l'état (facultatif). HAProxy Utilise par défaut la méthode HTTP OPTIONS. Vous pouvez aussi spécifier GET ou HEAD.
- Activer les statistiques (facultatif)
- Ports. Par défaut, AWS OpsWorks Stacks est configuré HAProxy pour gérer à la fois le trafic HTTP et HTTPS. Vous pouvez configurer HAProxy pour ne gérer que l'un ou l'autre en remplaçant le modèle de configuration Chef,`haproxy.cfg.erb`.

Recettes Setup :

- opsworks_initial_setup
- ssh_host_keys

- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- haproxy

Recettes Configure :

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- haproxy::configure

Recettes Deploy :

- deploy::default
- haproxy::configure

Recettes Shutdown :

- opsworks_shutdown::default
- haproxy::stop

Installation:

- AWS OpsWorks Stacks utilise le programme d'installation du package de l'instance pour HAProxy effectuer l'installation dans ses emplacements par défaut.
- Vous devez configurer syslog pour diriger les fichiers-journaux vers un emplacement spécifié. Pour de plus amples informations, veuillez consulter HAProxy.

HAProxy AWS OpsWorks Couche Stacks

> **Note**
>
> Cette couche est disponible uniquement pour Chef 11 et les piles antérieures basées sur Linux.

La couche AWS OpsWorks Stacks est une HAProxy couche AWS OpsWorks Stacks qui fournit un modèle pour les instances hébergeant un HAProxyserveur : un équilibre de charge fiable et performant. TCP/HTTP Une petite instance est généralement suffisante pour gérer tout le trafic du serveur d'application.

> **Note**
>
> Les piles sont limitées à une seule région. Pour distribuer votre application sur plusieurs régions, vous devez créer une pile distincte pour chaque région.

Pour créer une HAProxy couche

1. Dans le volet de navigation, cliquez sur Layers.
2. Sur la page Couches, cliquez sur Add a Layer (Ajouter une couche) ou + Layer. Pour Type de couche, sélectionnez HAProxy.

La couche a les paramètres de configuration suivants, qui sont tous facultatifs.

HAProxy statistiques

Indique si la couche de collecte et affiche les statistiques. La valeur par défaut est Yes (Oui).

URL des statistiques

Chemin d'URL de la page de statistiques. L'URL complète est http ://*DNSNameStatisticsPath*, où *DNSName* est le nom DNS de l'instance associée. La *StatisticsPath* valeur par défaut est /haproxy ? stats, ce qui correspond à quelque chose comme : http://ec2-54-245-151-7.us-west-2.compute.amazonaws.com/haproxy?stats.

Nom d'utilisateur des statistiques

Le nom d'utilisateur de la page de statistiques, que vous devez fournir pour afficher la page de statistiques. La valeur par défaut est « opsworks ».

Mot de passe des statistiques

Mot de passe de la page de statistiques que vous devez fournir pour consulter la page de statistiques. La valeur par défaut est une chaîne générée de façon aléatoire.

Health check URL

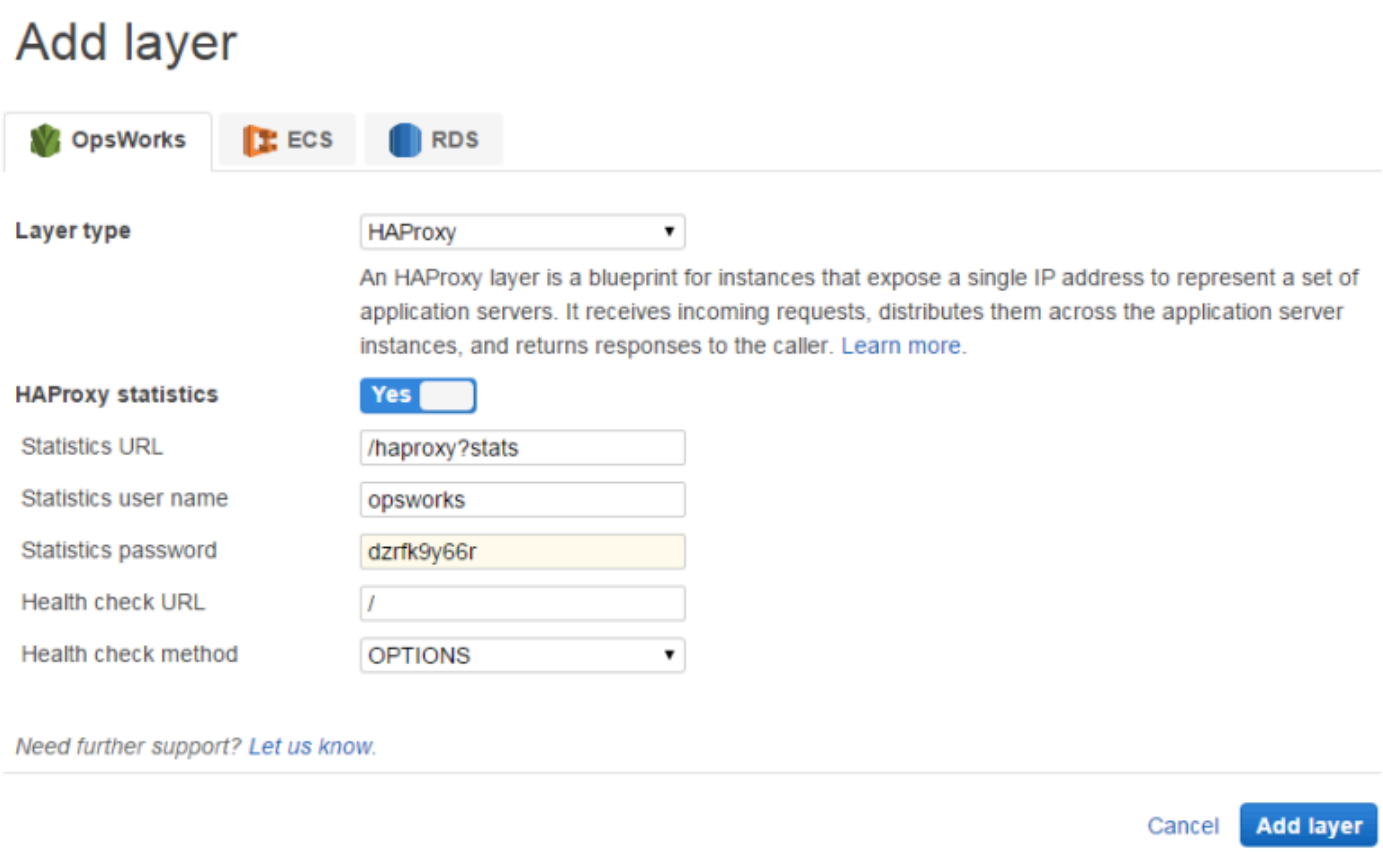
Suffixe de l'URL du bilan de santé. HAProxy utilise cette URL pour appeler périodiquement une méthode HTTP sur chaque instance de serveur d'applications afin de déterminer si l'instance fonctionne. Si le bilan de santé échoue, HAProxy arrête le routage du trafic vers l'instance jusqu'à ce qu'elle soit redémarrée, soit manuellement, soit par le biais d'une réparation automatique. La valeur par défaut du suffixe d'URL est «/», ce qui correspond à la page d'accueil de l'instance de serveur : http ://*DNSName*/.

Health check method

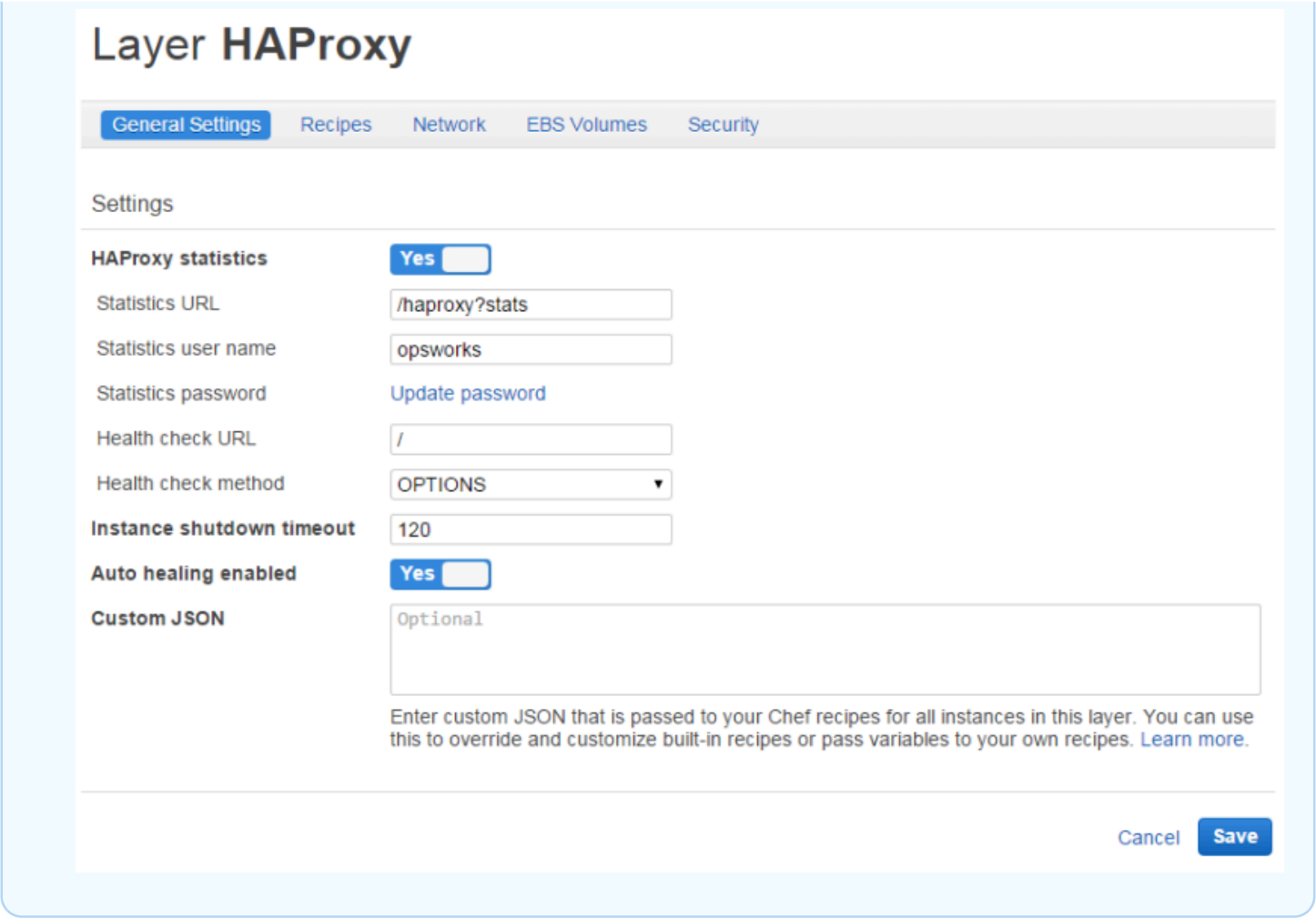
Méthode HTTP à utiliser pour vérifier si les instances fonctionnent. La valeur par défaut est OPTIONS et vous pouvez également spécifier GET ou HEAD. Pour plus d'informations, consultez httpchk.

Custom security groups

Ce paramètre apparaît si vous avez choisi de ne pas associer automatiquement un groupe de sécurité AWS OpsWorks Stacks intégré à vos couches. Vous devez spécifier le groupe de sécurité à associer à la couche. Assurez-vous que le groupe dispose des paramètres appropriés pour autoriser le trafic entre les couches. Pour de plus amples informations, veuillez consulter Créer une pile.



> **Note**
>
> Enregistrez le mot de passe pour une utilisation ultérieure ; AWS OpsWorks Stacks ne vous permet pas de voir le mot de passe après avoir créé la couche. Cependant, vous pouvez mettre à jour le mot de passe en accédant à la page Edit de la couche et en cliquant sur Update password sur l'onglet General Settings.



Fonctionnement de HAProxy la couche

Par défaut, HAProxy effectue les opérations suivantes :

- Il écoute les requêtes sur les ports HTTP et HTTPS.

  Vous pouvez configurer HAProxy pour écouter uniquement sur le port HTTP ou HTTPS en remplaçant le modèle de configuration Chef,`haproxy.cfg.erb`.
- Achemine le trafic entrant vers les instances qui sont membres de n'importe quelle couche de serveur d'application.

  Par défaut, AWS OpsWorks Stacks est configuré HAProxy pour distribuer le trafic aux instances membres de n'importe quelle couche de serveur d'applications. Vous pouvez, par exemple, avoir une pile contenant à la fois des couches Rails App Server et PHP App Server, et un HAProxy maître distribue le trafic aux instances des deux couches. Vous pouvez configurer le routage par défaut en utilisant une recette personnalisée.

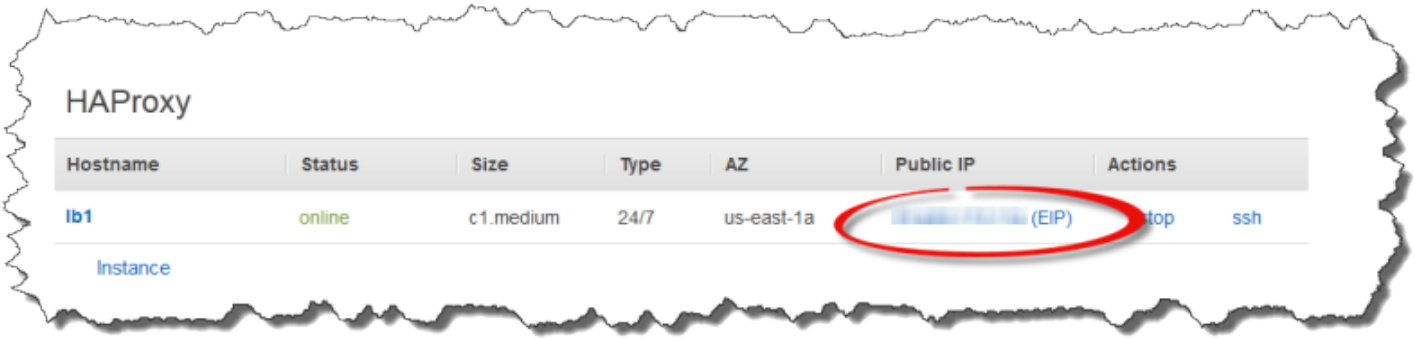
- Achemine le trafic sur plusieurs zones de disponibilité.

  Si une seule zone de disponibilité s'arrête, l'équilibreur de charge achemine le trafic entrant vers les instances dans d'autres zones de sorte que votre application continue à s'exécuter sans interruption. C'est pourquoi il est recommandé de distribuer vos serveurs d'applications sur plusieurs zones de disponibilité.
- Exécute périodiquement la méthode de vérification de l'état spécifiée sur chaque instance de serveur d'application afin d'évaluer son état.

  Si la méthode ne revient pas dans un délai spécifié, l'instance est présumée avoir échoué et HAProxy arrête le routage des demandes vers l'instance. AWS OpsWorks Stacks fournit également un moyen de remplacer automatiquement les instances défaillantes. Pour de plus amples informations, veuillez consulter Utilisation de la réparation automatique. Vous pouvez changer la méthode de vérification de l'état lorsque vous créez la couche.
- Collecte les statistiques et les affiche éventuellement sur une page web.

> **Important**
>
> Pour que la vérification de l'état fonctionne correctement avec la méthode OPTIONS par défaut, votre application doit retourner un code d'état 2xx ou 3xx.

Par défaut, lorsque vous ajoutez une instance à une HAProxy couche, AWS OpsWorks Stacks lui attribue une adresse IP élastique pour représenter l'application, qui est publique dans le monde entier. Dans la mesure où l'adresse IP Elastic de l'instance HAProxy est la seule URL publiquement exposée de l'application, vous n'avez pas à créer ni à gérer les noms de domaine public pour les instances de serveur d'application sous-jacentes. Vous pouvez obtenir l'adresse en accédant à la page Instances et en examinant l'adresse IP publique de l'instance, comme le montre l'illustration suivante. Une adresse suivie d'(EIP) est une adresse IP Elastic. Pour plus d'informations sur les adresses IP Elastic, consultez Adresses IP Elastic (EIP).



Lorsque vous arrêtez une HAProxy instance, AWS OpsWorks Stacks conserve l'adresse IP élastique et la réaffecte à l'instance lorsque vous la redémarrez. Si vous supprimez une HAProxy instance, AWS OpsWorks Stacks supprime par défaut l'adresse IP de l'instance. Pour conserver l'adresse, désactivez l'option Supprimer l'adresse IP Elastic de l'instance, comme indiqué dans l'illustration suivante.



Cette option affecte ce qui se passe lorsque vous ajoutez une instance à la couche pour remplacer une instance supprimée :

- Si vous avez conservé l'adresse IP élastique de l'instance supprimée, AWS OpsWorks Stacks attribue l'adresse à la nouvelle instance.
- Dans le cas contraire, AWS OpsWorks Stacks attribue une nouvelle adresse IP élastique à l'instance et vous devez mettre à jour les paramètres de votre bureau d'enregistrement DNS pour qu'ils soient mappés à la nouvelle adresse.

Lorsque des instances de serveur d'applications sont mises en ligne ou hors ligne, manuellement ou à la suite d'un dimensionnement automatique ou d'une réparation automatique, la configuration de

l'équilibreur de charge doit être mise à jour pour acheminer le trafic vers l'ensemble actuel d'instances en ligne. Cette tâche est gérée automatiquement par les recettes intégrées de la couche :

- Lorsque de nouvelles instances sont mises en ligne, AWS OpsWorks Stacks déclenche un événement de configuration du cycle de vie. Les recettes de configuration intégrées de la HAProxy couche mettent à jour la configuration de l'équilibreur de charge afin qu'il distribue également les demandes à toutes les nouvelles instances de serveur d'applications.
- Lorsque des instances sont hors ligne ou qu'une instance échoue à un contrôle de santé, AWS OpsWorks Stacks déclenche également un événement de configuration du cycle de vie. Les recettes de HAProxy configuration mettent à jour la configuration de l'équilibreur de charge pour acheminer le trafic uniquement vers les instances en ligne restantes.

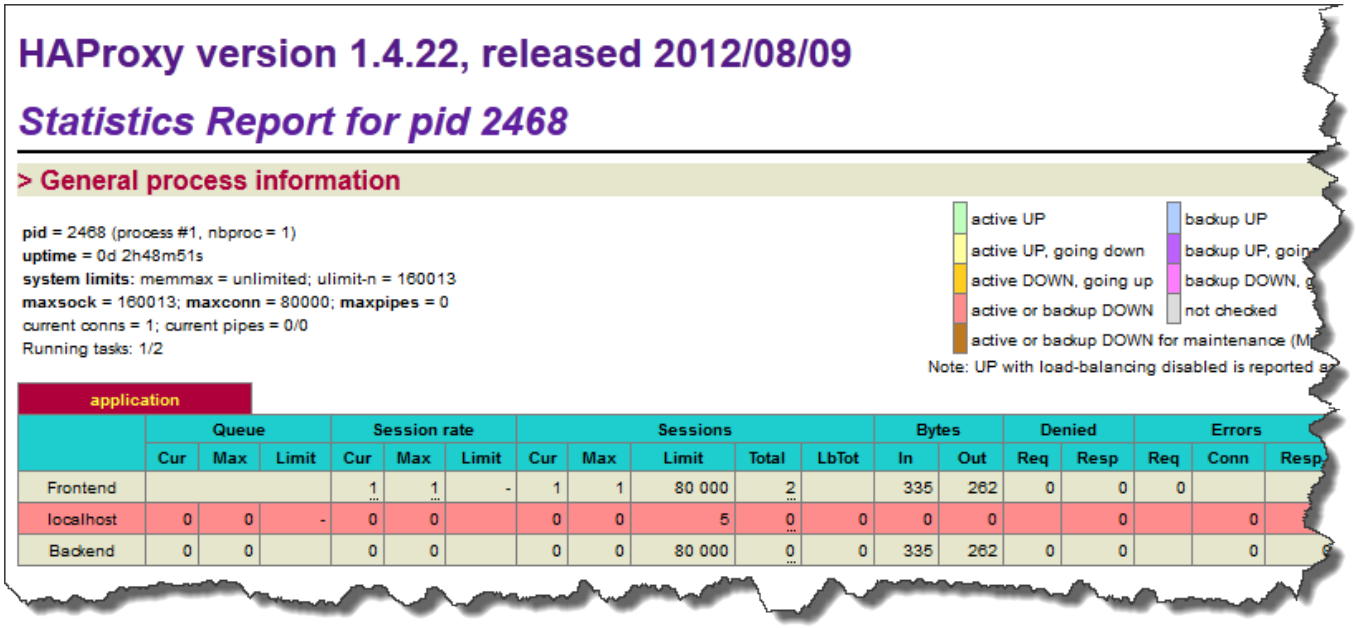
Enfin, vous pouvez également utiliser un domaine personnalisé avec la HAProxy couche. Pour de plus amples informations, veuillez consulter Utilisation des domaines personnalisés.

Page de statistiques

Si vous avez activé la page de statistiques, elle HAProxy affiche une page contenant diverses mesures à l'URL spécifiée.

Pour consulter les HAProxy statistiques

1. Obtenez le HAProxy nom DNS public de l'instance sur la page Détails de l'instance et copiez-le.
2. Sur la page Couches, cliquez sur HAProxy pour ouvrir la page des détails de la couche.
3. Obtenez l'URL des statistiques à partir des détails de la couche et ajoutez-la au nom DNS public. Par exemple : **`http://ec2-54-245-102-172.us-west-2.compute.amazonaws.com/haproxy?stats`**.
4. Collez l'URL de l'étape précédente dans votre navigateur et utilisez le nom d'utilisateur et le mot de passe que vous avez spécifiés lorsque vous avez créé la couche pour ouvrir la page de statistiques.



Référence de couche MySQL

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour les piles Linux.

La couche MySQL prend en charge MySQL, un système de gestion de base de données relationnelle largement utilisé. AWS OpsWorks Stacks installe la version la plus récente disponible, qui dépend du système d'exploitation. Si vous ajoutez une instance MySQL, les informations d'accès nécessaires sont fournies aux couches serveurs d'applications. Vous devez écrire des recettes Chef personnalisées pour configurer des configurations maître-maître ou maître-esclave.

Nom court : db-master

Compatibilité : une couche MySQL est compatible avec les couches suivantes : custom, lb, memcached, monitoring-master, nodejs-app, php-app, rails-app et web.

Ports ouverts : une couche MySQL permet l'accès public au port 22 (SSH) et à tous les ports depuis les serveurs Web, les serveurs personnalisés et les serveurs d'applications Rails, PHP et Node.js de la pile.

Autoassign Elastic IP addresses : Désactivé par défaut

Default EBS volume (Volume EBS par défaut) : Oui, dans `/vol/mysql`

Groupe de sécurité par défaut : AWS-OpsWorks-DB-Master -Server

Configuration : Pour configurer une couche MySQL, vous devez spécifier les éléments suivants :

- Mot de passe utilisateur racine
- Moteur MySQL

Recettes Setup :

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- mysql::server
- dependencies
- deploy::mysql

Recettes Configure :

- opsworks_ganglia::configure-client

- ssh_users
- agent_version
- deploy::mysql

Recettes Deploy :

- deploy::default
- deploy::mysql

Recettes Shutdown :

- opsworks_shutdown::default
- mysql::stop

Installation:

- AWS OpsWorks Stacks utilise le programme d'installation du package de l'instance pour installer MySQL et ses fichiers journaux à leur emplacement par défaut. Pour plus d'informations, consultez Documentation MySQL.

OpsWorks Couche MySQL

Important

Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Note

Cette couche est disponible uniquement pour Chef 11 et les piles antérieures basées sur Linux.

Une OpsWorks couche MySQL fournit un modèle pour les EC2 instances Amazon qui fonctionnent comme un maître de base de données MySQL. Une recette intégrée crée une base de données pour chaque application qui a été déployée sur une couche du serveur d'applications. Par exemple, si vous déployez une application PHP « myapp », la recette crée une base de données « myapp ».

La couche MySQL possède les paramètres de configuration suivants.

Mot de passe utilisateur racine MySQL

(Obligatoire) Le mot de passe utilisateur racine.

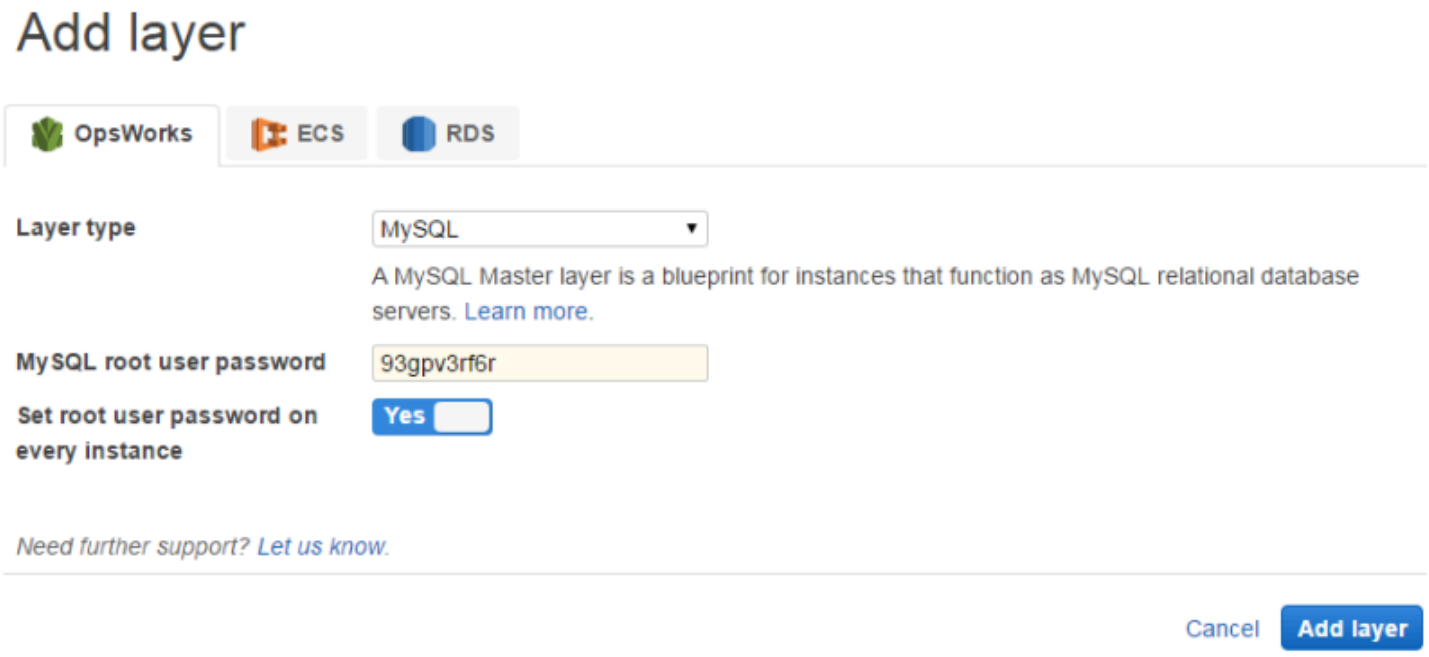
Définir le mot de passe utilisateur racine sur chaque instance

(Facultatif) Indique si le mot de passe utilisateur racine est inclus dans les attributs de configuration et de déploiement de la pile qui sont installés sur chaque instance de la pile. Le paramètre par défaut est Yes.

Si vous définissez cette valeur sur Non, AWS OpsWorks Stacks transmet le mot de passe root uniquement aux instances du serveur d'applications.

Custom security groups

(Facultatif) Un groupe de sécurité personnalisés à associer à la couche. Pour de plus amples informations, veuillez consulter Créer une pile.



Vous pouvez ajouter une ou plusieurs instances à la couche, chacune d'entre elles représentant une base de données principale MySQL distincte. Vous pouvez ensuite attacher une instance à une application, qui installe les informations de connexion nécessaires sur les serveurs d'applications de l'application. L'application peut ensuite utiliser les informations de connexion pour se connecter au serveur de base de données de l'instance.

Référence des couches serveur d'applications

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks prend en charge plusieurs serveurs d'applications et de pages Web statiques.

Rubriques

- Référence de la couche AWS Flow (Ruby)
- Référence de la couche Java App Server
- Référence de la couche de serveur d'applications Node.js
- Référence de la couche PHP App Server
- Référence de la couche Rails App Server
- Référence de couche de serveur Web statique

Référence de la couche AWS Flow (Ruby)

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des

questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour les piles Linux.

Une couche AWS Flow (Ruby) fournit un modèle pour les instances qui hébergent l'activité d'Amazon Simple Workflow Service et les travailleurs du flux de travail.

Nom abrégé : aws-flow-ruby

Compatibilité : une couche AWS Flow (Ruby) est compatible avec PHP App Server, MySQL, Memcached, Ganglia et les couches personnalisées.

Ports ouverts : Aucun.

Rôle IAM : aws-opsworks-ec 2- role-with-swf est le rôle AWS Flow (Ruby) standard que AWS OpsWorks Stacks crée pour vous, sur demande.

Autoassign Elastic IP addresses : Désactivé par défaut

Default EBS Volume (Volume EBS par défaut) : Non

Groupe de sécurité par défaut : AWS-OpsWorks-AWS-Flow -Ruby-Server

Recettes Setup :

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- opsworks_aws_flow_ruby::setup

Recettes Configure :

- opsworks_ganglia::configure-client
- ssh_users
- mysql::client
- agent_version
- opsworks_aws_flow_ruby::configure

Recettes Deploy :

- deploy::default
- déployer : aws-flow-ruby

Recettes Undeploy :

- déployer : aws-flow-ruby-undeploy

Recettes Shutdown :

- opsworks_shutdown::default

Référence de la couche Java App Server

Important

Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Note

Cette couche est disponible uniquement pour les piles Linux.

Une couche Java App Server prend en charge un serveur d'applications Apache Tomcat 7.0.

Nom court : java-app

Compatibilité : une couche Java App Server est compatible avec les couches suivantes : custom, db-master et memcached.

Ports ouverts : une couche Java App Server permet l'accès public aux ports 22 (SSH), 80 (HTTP), 443 (HTTPS) et à tous les ports des équilibreurs de charge.

Autoassign Elastic IP addresses : Désactivé par défaut

Default EBS Volume (Volume EBS par défaut) : Non

Groupe de sécurité par défaut : AWS-OpsWorks-Java-App -Server

Recettes Setup :

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- opsworks_java::setup

Recettes Configure :

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- opsworks_java::configure

Recettes Deploy :

- deploy::default
- deploy::java

Recettes Undeploy :

- deploy::java-undeploy

Recettes Shutdown :

- opsworks_shutdown::default
- deploy::java-stop

Installation:

- Tomcat s'installe dans `/usr/share/tomcat7`.
- Pour plus d'informations sur la façon de produire des fichiers journaux, consultez Journalisation dans Tomcat.

Référence de la couche de serveur d'applications Node.js

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour les piles Linux.

Une couche de serveur d'applications Node.js prend en charge un serveur d'applications Node.js, qui est une plate-forme permettant de mettre en œuvre des serveurs d'applications réseau hautement évolutifs. Les programmes sont écrits en JavaScript utilisant le mode asynchrone piloté par les événements afin de minimiser les frais généraux et I/O de maximiser l'évolutivité.

Nom court : nodejs-app

Compatibilité : une couche de serveur d'applications Node.js est compatible avec les couches suivantes : custom, db-master, memcached et monitoring-master.

Ports ouverts : une couche de serveur d'applications Node.js permet l'accès public aux ports 22 (SSH), 80 (HTTP), 443 (HTTPS) et à tous les ports des équilibreurs de charge.

Autoassign Elastic IP addresses : Désactivé par défaut

Default EBS volume : Non

Groupe de sécurité par défaut : AWS-OpsWorks-nodejs-App -Server

Recettes Setup :

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- opsworks_nodejs
- opsworks_nodejs::npm

Recettes Configure :

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- opsworks_nodejs::configure

Recettes Deploy :

- deploy::default
- opsworks_nodejs
- opsworks_nodejs::npm

- deploy::nodejs

Recettes Undeploy :

- deploy::nodejs-undeploy

Recettes Shutdown :

- opsworks_shutdown::default
- deploy::nodejs-stop

Installation:

- Node.js s'installe dans `/usr/local/bin/node`.
- Pour plus d'informations sur la façon de produire les fichiers journaux, consultez Comment journaliser dans node.js sur le site web Nodejitsu.

Configuration de l'application node.js :

- Le fichier principal géré par Node.js doit être nommé `server.js` et résider dans le répertoire racine de l'application déployée.
- L'application Node.js doit être définie pour écouter sur le port 80 (ou le port 443, le cas échéant).

> **Note**
>
> Les applications Node.js qui exécutent Express utilisent généralement le code suivant pour définir le port d'écoute, où `process.env.PORT` représente le port par défaut et se résout en port 80 :
>
> ```
> app.set('port', process.env.PORT || 3000);
> ```
>
> Avec AWS OpsWorks Stacks, vous devez spécifier explicitement le port 80, comme suit :
>
> ```
> app.set('port', 80);
> ```

Référence de la couche PHP App Server

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour les piles Linux.

La couche PHP App Server prend en charge un serveur d'applications PHP en utilisant Apache2 avec mod_php.

Nom court : php-app

Compatibilité : une couche PHP App Server est compatible avec les couches suivantes : custom, db-master, memcached, monitoring-master et rails-app.

Ports ouverts : une couche PHP App Server permet l'accès public aux ports 22 (SSH), 80 (HTTP), 443 (HTTPS) et à tous les ports des équilibreurs de charge.

Autoassign Elastic IP addresses : Désactivé par défaut

Default EBS volume : Non

Groupe de sécurité par défaut : AWS-OpsWorks-PHP-App -Server

Recettes Setup :

- opsworks_initial_setup
- ssh_host_keys
- ssh_users

- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- mysql::client
- dependencies
- mod_php5_apache2

Recettes Configure :

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- mod_php5_apache2::php
- php::configure

Recettes Deploy :

- deploy::default
- deploy::php

Recettes Undeploy :

- deploy::php-undeploy

Recettes Shutdown :

- opsworks_shutdown::default
- apache2::stop

Installation :

- AWS OpsWorks Stacks utilise le programme d'installation du package de l'instance pour installer Apache2, mod_php et les fichiers journaux associés dans leurs emplacements par défaut.

Pour plus d'informations sur l'installation, consultez Apache. Pour plus d'informations sur la journalisation, consultez Fichiers journaux.

Référence de la couche Rails App Server

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour les piles Linux.

La couche Rails App Server prend en charge un serveur d'applications Ruby on Rails.

Nom court : rails-app

Compatibilité : une couche Rails App Server est compatible avec les couches suivantes : custom, db-master, memcached, monitoring-master, php-app.

Ports : une couche Rails App Server permet l'accès public aux ports 22 (SSH), 80 (HTTP), 443 (HTTPS) et à tous les ports des équilibreurs de charge.

Autoassign Elastic IP addresses : Désactivé par défaut

Default EBS volume : Non

Groupe de sécurité par défaut : AWS-OpsWorks-Rails-App -Server

Configuration : Pour configurer une couche Rails App Server, vous devez spécifier les éléments suivants :

- Ruby version

- Rails stack
- Rubygems version
- Indique s'il convient d'installer et de gérer Bundler
- Bundler version

Recettes Setup :

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- apache2 apache2::mod_deflate
- passenger_apache2
- passenger_apache2::mod_rails
- passenger_apache2::rails

Recettes Configure :

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- rails::configure

Recettes Deploy :

- deploy::default
- deploy::rails

Recettes Undeploy :

- deploy::rails-undeploy

Recettes Shutdown :

- opsworks_shutdown::default
- apache2::stop

Installation:

- AWS OpsWorks Stacks utilise le programme d'installation du package de l'instance pour installer Apache2 avec mod_passenger, mod_rails et les fichiers journaux associés à leurs emplacements par défaut. Pour plus d'informations sur l'installation, consultez la page Phusion Passenger. Pour plus d'informations sur la journalisation, consultez Fichiers journaux.

Référence de couche de serveur Web statique

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour les piles Linux.

La couche Static Web Server diffuse des pages HTML statiques, qui peuvent inclure du code côté client tel que. JavaScript Elle s'appuie sur Nginx, qui est un serveur HTTP open source, un serveur proxy inverse et un serveur proxy de messagerie.

Short name : web

Compatibilité : une couche de serveur Web statique est compatible avec les couches suivantes : custom, db-master, memcached.

Ports ouverts : une couche de serveur Web statique permet au public d'accéder aux ports 22 (SSH), 80 (HTTP), 443 (HTTPS) et à tous les ports des équilibreurs de charge.

Autoassign Elastic IP addresses : Désactivé par défaut

Default EBS volume : Non

Groupe de sécurité par défaut : AWS-OpsWorks-Web-Server

Recettes Setup :

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- nginx

Recettes Configure :

- opsworks_ganglia::configure-client
- ssh_users
- agent_version

Recettes Deploy :

- deploy::default
- deploy::web

Recettes Undeploy :

- deploy::web-undeploy

Recettes Shutdown :

- opsworks_shutdown::default
- nginx::stop

Installation:

- Nginx s'installe dans `/usr/sbin/nginx`.
- Les fichiers journaux Nginx sont dans `/var/log/nginx`.

Couches serveur d'applications

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Ces couches sont disponibles uniquement pour les piles Linux de Chef 11 et des versions antérieures.

AWS OpsWorks Stacks prend en charge plusieurs serveurs d'applications différents, où le terme « application » inclut les pages Web statiques. Chaque type de serveur possède une couche AWS OpsWorks Stacks distincte, avec des recettes intégrées qui gèrent l'installation du serveur d'applications et de tous les packages associés sur chacune des instances de la couche, le déploiement des applications, etc. Par exemple, la couche Java App Server installe plusieurs packages, notamment Apache, Tomcat et OpenJDK, et déploie des applications Java sur chacune des instances de la couche.

Voici la procédure de base pour l'utilisation des couches serveur d'application :

1. Créez l'un des types de couche App Server disponibles.
2. Ajoutez une ou plusieurs instances à la couche.

3. Créez les applications et déployez-les sur les instances. Pour de plus amples informations, veuillez consulter Applications.
4. (Facultatif) Si la couche comporte plusieurs instances, vous pouvez ajouter un équilibreur de charge, qui répartit le trafic entrant entre les instances. Pour de plus amples informations, veuillez consulter HAProxy AWS OpsWorks Couche Stacks.

Rubriques

- Couche AWS Flow (Ruby)
- Java App Server AWS OpsWorks Stacks Layer
- Node.js App Server AWS OpsWorks Stacks Layer
- couche d' AWS OpsWorks empilements de serveurs d'applications PHP
- Rails App Server AWS OpsWorks Stacks Layer
- Couche Static Web Server AWS OpsWorks Stacks

Couche AWS Flow (Ruby)

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour les piles Linux.

Une couche AWS Flow (Ruby) est une couche AWS OpsWorks Stacks qui fournit un plan pour les instances hébergeant l'activité Amazon SWF et les travailleurs du flux de travail. Les travailleurs sont mis en œuvre à l'aide de l'AWS Flow Framework for Ruby, un framework de programmation qui simplifie le processus de mise en œuvre d'une application asynchrone distribuée tout en offrant tous les avantages d'Amazon SWF. Cette infrastructure est idéale pour implémenter des applications et

satisfaire une large gamme de scénarios, y compris les processus métier, l'encodage du média, les tâches de longue durée et le traitement en arrière-plan.

La couche AWS Flow (Ruby) inclut les paramètres de configuration suivants.

RubyGems version

Version Gem de l'infrastructure.

Bundler version (Version de Bundler)

Version de Bundler.

EC2 Profil de l'instance

Un profil d' EC2 instance Amazon défini par l'utilisateur à utiliser par les instances de la couche. Ce profil doit autoriser les applications exécutées sur les instances de la couche à accéder à Amazon SWF.

Si votre compte ne possède pas de profil approprié, vous pouvez sélectionner Nouveau profil avec accès SWF pour que AWS OpsWorks Stacks mette à jour le profil ou vous pouvez le mettre à jour vous-même à l'aide de la console IAM. Vous pouvez ensuite utiliser le profil mis à jour pour toutes les couches AWS Flow suivantes. Voici une brève description de la procédure à suivre pour créer le profil à l'aide de la console IAM. Pour plus d'informations, consultez Identity and Access Management dans Amazon Simple Workflow Service.

Création d'un profil pour les instances AWS Flow (Ruby)

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Choisissez Politiques dans le volet de navigation, puis choisissez Créer une politique pour créer une nouvelle politique gérée par le client.
3. Pour Service, choisissez SWF.
4. Pour Actions, sélectionnez Toutes les actions SWF (swf : *).
5. Pour Amazon Resource Name (ARN), entrez l'ARN qui spécifie les domaines Amazon SWF auxquels les utilisateurs peuvent accéder. Choisissez **`All resources`** de donner accès à tous les domaines.
6. Choisissez Suivant.
7. Entrez éventuellement une balise pour identifier la politique.

8. Choisissez Suivant.
9. Lorsque vous avez terminé, choisissez Create policy.
10. Choisissez Rôles dans le volet de navigation, puis choisissez Créer un rôle.
11. Spécifiez le nom du rôle et choisissez Next Step. Vous ne pouvez pas modifier le nom après que le rôle a été créé.
12. Choisissez le service AWS, puis choisissez EC2.
13. Choisissez Suivant.
14. Dans la liste des politiques d'autorisation, choisissez la politique que vous avez créée précédemment.
15. Choisissez Suivant.
16. Saisissez un nom de rôle, puis choisissez Create role (Créer un rôle). Vous ne pouvez pas modifier le nom après que le rôle a été créé.
17. Spécifiez ce profil lorsque vous créez une couche AWS Flow (Ruby) dans AWS OpsWorks Stacks.

Java App Server AWS OpsWorks Stacks Layer

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour les piles Linux.

La couche Java App Server est une couche AWS OpsWorks Stacks qui fournit un modèle pour les instances qui fonctionnent comme des serveurs d'applications Java. Cette couche est basée sur Apache Tomcat 7.0 et Open JDK 7. AWS OpsWorks Stacks installe également la bibliothèque de

connecteurs Java, qui permet aux applications Java d'utiliser un `DataSource` objet JDBC pour se connecter à un magasin de données principal.

Installation : Tomcat est installé dans `/usr/share/tomcat7`.

La page Add Layer (Ajouter une couche) fournit les options de configuration suivantes :

Java VM Options (Options de machine virtuelle Java)

Vous pouvez utiliser ce paramètre pour spécifier les options de machine virtuelle Java personnalisées ; il n'y a pas d'options par défaut. Par exemple, un ensemble commun d'options est `-Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC`. Si vous utilisez Java VM Options, assurez-vous de transmettre un ensemble d'options valide ; AWS OpsWorks Stacks ne valide pas la chaîne. Si vous essayez de passer une option non valide, le serveur Tomcat ne peut généralement pas démarrer, ce qui entraîne un échec de l'installation. Si cela se produit, vous pouvez examiner le journal Chef de l'installation de l'instance pour obtenir plus de détails. Pour plus d'informations sur la façon d'afficher et d'interpréter les journaux Chef, consultez Journaux de Chef.

Custom security groups

Ce paramètre apparaît si vous avez choisi de ne pas associer automatiquement un groupe de sécurité AWS OpsWorks Stacks intégré à vos couches. Vous devez spécifier le groupe de sécurité à associer à la couche. Pour de plus amples informations, veuillez consulter Créer une pile.

Elastic Load Balancer

Vous pouvez associer un équilibreur de charge Elastic Load Balancing aux instances de la couche. Pour de plus amples informations, veuillez consulter Couche Elastic Load Balancing.

Vous pouvez spécifier d'autres paramètres de configuration à l'aide d'un JSON personnalisé ou d'un fichier d'attributs personnalisé. Pour de plus amples informations, veuillez consulter Configuration personnalisée.

> **Important**
>
> Si votre application Java utilise le protocole SSL, nous vous recommandons de le désactiver SSLv3 si possible pour corriger les vulnérabilités décrites dans CVE-2014-3566. Pour de plus amples informations, veuillez consulter Désactivation SSLv3 pour les serveurs Apache.

Rubriques

- Désactivation SSLv3 pour les serveurs Apache
- Configuration personnalisée
- Déploiement des applications Java

Désactivation SSLv3 pour les serveurs Apache

Pour le désactiver SSLv3, vous devez modifier les `SSLProtocol` paramètres du `ssl.conf` fichier du serveur Apache. Pour ce faire, vous devez remplacer le fichier `ssl.conf.erb` modèle du livre de recettes Apache2 intégré, que les recettes de configuration de la couche Java App Server utilisent pour créer. `ssl.conf` Les détails dépendent du système d'exploitation que vous spécifiez pour les instances de la couche. Ce qui suit récapitule les modifications requises pour les systèmes Amazon Linux et Ubuntu. SSLv3 est automatiquement désactivé pour les systèmes Red Hat Enterprise Linux (RHEL). Pour plus d'informations sur le remplacement d'un modèle intégré, consultez Utilisation de modèles personnalisés.

Amazon Linux

Le fichier `ssl.conf.erb` pour ces systèmes d'exploitation figure dans le répertoire `apache2` du livre de recettes `apache2/templates/default/mods`. L'exemple suivant montre la partie pertinente du fichier intégré.

```
...
#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>
```

Remplacez `ssl.conf.erb` et modifiez le paramètre `SSLProtocol` comme suit.

```
...
#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

Le fichier `ssl.conf.erb` pour ce système d'exploitation figure dans le répertoire `apache2` du livre de recettes `apache2/templates/ubuntu-14.04/mods`. L'exemple suivant montre la partie pertinente du fichier intégré.

```
...
# The protocols to enable.
# Available values: all, SSLv3, TLSv1.2
# SSL v2 is no longer supported
SSLProtocol all
...
```

Modifiez ce paramètre en ce qui suit.

```
...
# The protocols to enable.
# Available values: all, SSLv3, TLSv1.2
# SSL v2 is no longer supported
SSLProtocol all -SSLv3 -SSLv2
...
```

Configuration personnalisée

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks expose des paramètres de configuration supplémentaires sous forme d'attributs intégrés, qui se trouvent tous dans l'`opsworks_java`espace de noms. Vous pouvez

utiliser un JSON personnalisé ou un fichier d'attributs personnalisé pour remplacer les attributs intégrés et spécifier des valeurs personnalisées. Par exemple, les versions de la machine virtuelle Java et de Tomcat sont respectivement représentées par les attributs intégrés `jvm_version` et `java_app_server_version`, les deux ayant la valeur 7. Vous pouvez utiliser un JSON personnalisé ou un fichier d'attributs personnalisé pour définir l'une et/ou l'autre avec la valeur 6. L'exemple suivant utilise JSON personnalisé pour définir les deux attributs avec la valeur 6 :

```
{
  "opsworks_java": {
    "jvm_version": 6,
    "java_app_server_version" : 6
  }
}
```

Pour de plus amples informations, veuillez consulter Utilisation du JSON personnalisé.

Un autre exemple de configuration personnalisée consiste à installer un JDK personnalisé en remplaçant les attributs `use_custom_pkg_location`, `custom_pkg_location_url_debian` et `custom_pkg_location_url_rhel`.

> **Note**
>
> Si vous remplacez les livres de recettes intégrés, vous devez mettre à jour ces composants vous-même.

Pour de plus amples informations sur les attributs et leur remplacement, consultez Remplacement des attributs. Pour obtenir la liste des attributs intégrés, consultez Attributs opsworks_java.

Déploiement des applications Java

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les rubriques suivantes décrivent comment déployer des applications sur les instances d'une couche Java App Server. Les exemples concernent des applications JSP, mais vous pouvez, pour l'essentiel, utiliser la même procédure pour installer d'autres types d'applications Java.

Vous pouvez déployer des pages JSP à partir de n'importe lequel des référentiels pris en charge. Si vous souhaitez déployer des fichiers WAR, notez que AWS OpsWorks Stacks extrait automatiquement les fichiers WAR déployés depuis une archive Amazon S3 ou HTTP, mais pas depuis un dépôt Git ou Subversion. Si vous souhaitez utiliser Git ou Subversion pour les fichiers WAR, vous pouvez effectuer l'une des actions suivantes :

- Stocker l'archive extraite dans le référentiel.
- Stocker le fichier WAR dans le référentiel et utiliser un hook de déploiement Chef pour extraire l'archive, comme décrit dans l'exemple suivant.

Vous pouvez utiliser les raccordements de déploiement Chef pour exécuter les applications Ruby fournies par l'utilisateur sur une instance à n'importe laquelle des quatre étapes du déploiement. Le nom de l'application détermine l'étape. Voici un exemple d'application Ruby nommée `before_migrate.rb`, qui extrait un fichier WAR déployé à partir d'un référentiel Git ou Subversion. Comme le nom associe l'application au hook de déploiement Checkout, il s'exécute au début de l'opération de déploiement, une fois que le code a été vérifié, mais avant la migration. Pour plus d'informations sur l'utilisation de cet exemple, consultez Utilisation des raccordements de déploiement Chef.

```
::Dir.glob(::File.join(release_path, '*.war')) do |archive_file|
  execute "unzip_#{archive_file}" do
    command "unzip #{archive_file}"
    cwd release_path
  end
end
```

> **Note**
>
> Lorsque vous déployez une mise à jour sur une application JSP, il est possible que Tomcat ne reconnaisse pas la mise à jour et continue à exécuter la version d'application existante. Cela peut se produire, par exemple, si vous déployez votre application en tant que fichier .zip qui contient uniquement une page JSP. Afin de garantir que Tomcat exécute la version la plus récemment déployée, le répertoire racine du projet doit inclure un répertoire WEB-INF

qui contient un fichier `web.xml`. Un fichier `web.xml` peut contenir une grande variété de contenus, mais ce qui suit est suffisant pour veiller à ce que Tomcat reconnaisse les mises à jour et exécute la version d'application actuellement déployée. Vous n'avez pas à changer la version pour chaque mise à jour. Tomcat reconnaît la mise à jour même si la version n'a pas changé.

```
<context-param>
  <param-name>appVersion</param-name>
  <param-value>0.1</param-value>
</context-param>
```

Rubriques

- Déploiement d'une application JSP
- Déploiement d'une application JSP avec une base de données principale

Déploiement d'une application JSP

Pour déployer une application JSP, spécifiez le nom et le référentiel. Vous pouvez aussi, le cas échéant, spécifier des domaines et des paramètres SSL. Pour plus d'informations sur la création d'une application, consultez Ajout d'applications. La procédure suivante explique comment créer et déployer une page JSP simple à partir d'une archive Amazon S3 publique. Pour plus d'informations sur l'utilisation d'autres types de référentiels, y compris les archives privées Amazon S3, consultezSource de l'application.

L'exemple suivant montre la page JSP, qui affiche simplement quelques informations système.

```
<%@ page import="java.net.InetAddress" %>
<html>
<body>
<%
    java.util.Date date = new java.util.Date();
    InetAddress inetAddress = InetAddress.getLocalHost();
%>
The time is
<%
    out.println( date );
```

```
    out.println("<br>Your server's hostname is "+inetAddress.getHostName());
%>
<br>
</body>
</html>
```

> **Note**
>
> La procédure suivante suppose que vous maîtrisiez déjà les bases de la création de piles, d'ajout d'instances aux couches, et ainsi de suite. Si vous utilisez AWS OpsWorks Stacks pour la première fois, vous devriez d'abord voirMise en route des piles Linux Chef 11.

Pour déployer une page JSP à partir d'une archive Amazon S3

1. Créez une pile avec une couche Java App Server, ajoutez une instance 24 h/24 et 7 j/7 à la couche et démarrez-la.
2. Copiez le code dans un fichier nommé `simplejsp.jsp`, placez le fichier dans un dossier nommé `simplejsp` et créez une archive `.zip` du dossier. Les noms sont arbitraires ; vous pouvez utiliser n'importe quels noms de fichier ou de dossier de votre choix. Vous pouvez aussi utiliser d'autres types d'archive, y compris gzip, bzip2, tarball ou Java WAR. Notez que AWS OpsWorks Stacks ne prend pas en charge les archives non compressées. Pour déployer plusieurs pages JSP, incluez-les dans la même archive.
3. Téléchargez l'archive dans un compartiment Amazon S3 et rendez le fichier public. Copiez l'URL du fichier pour une utilisation ultérieure. Pour plus d'informations sur la création de compartiments et le chargement des fichiers, consultez Mise en route avec Amazon Simple Storage Service.
4. Ajoutez une application à la pile et spécifiez les paramètres suivants :

   - Nom – `SimpleJSP`
   - Type d'application — `Java`
   - Type de référentiel — `Http Archive`
   - URL du référentiel : URL Amazon S3 de votre fichier d'archive.

   Utilisez les valeurs par défaut pour les paramètres restants, puis cliquez sur Ajouter une application pour créer l'application.

5. Déployez l'application sur l'instance Java App Server.

Vous pouvez désormais accéder à l'URL de l'application et afficher l'application. Si vous n'avez pas spécifié de domaine, vous pouvez créer une URL en utilisant l'adresse IP publique de l'instance ou son nom DNS public. Pour obtenir l'adresse IP publique ou le nom DNS public d'une instance, accédez à la console AWS OpsWorks Stacks et cliquez sur le nom de l'instance sur la page Instances pour ouvrir sa page de détails.

Le reste de l'URL dépend du nom abrégé de l'application, qui est un nom en minuscules généré par AWS OpsWorks Stacks à partir du nom de l'application que vous avez spécifié lors de sa création. Par exemple, le nom court de SimpleJSP est simplejsp. Vous pouvez obtenir le nom court d'une application à partir de la page des détails.

- Si le nom court est `root`, vous pouvez utiliser `http://`*`public_DNS`*`/`*`appname`*`.jsp` ou `http://`*`public_IP`*`/`*`appname`*`.jsp`.
- Sinon, vous pouvez utiliser `http://`*`public_DNS`*`/`*`app_shortname`*`/`*`appname`*`.jsp` ou `http://`*`public_IP`*`/`*`app_shortname`*`/`*`appname`*`.jsp`.

Si vous avez spécifié un domaine pour l'application, l'URL est `http://`*`domain`*`/`*`appname`*`.jsp`.

L'URL de l'exemple doit ressembler à quelque chose comme `http://192.0.2.0/simplejsp/simplejsp.jsp`.

Si vous souhaitez déployer plusieurs applications sur la même instance, vous ne devez pas utiliser `root` comme nom court. Il peut s'ensuivre des conflits d'URL qui empêchent que l'application ne fonctionne correctement. Au lieu de cela, attribuez un nom de domaine différent à chaque application.

Déploiement d'une application JSP avec une base de données principale

Les pages JSP peuvent utiliser un objet JDBC `DataSource` pour se connecter à une base de données principale. Vous créez et déployez une telle application à l'aide de la procédure de la section précédente, avec une étape supplémentaire pour configurer la connexion.

La page JSP suivante montre comment se connecter à un objet `DataSource`.

```
<html>
  <head>
    <title>DB Access</title>
  </head>
```

```
<body>
  <%@ page language="java" import="java.sql.*,javax.naming.*,javax.sql.*" %>
  <%
    StringBuffer output = new StringBuffer();
    DataSource ds = null;
    Connection con = null;
    Statement stmt = null;
    ResultSet rs = null;
    try {
      Context initCtx = new InitialContext();
      ds = (DataSource) initCtx.lookup("java:comp/env/jdbc/mydb");
      con = ds.getConnection();
      output.append("Databases found:<br>");
      stmt = con.createStatement();
      rs = stmt.executeQuery("show databases");
      while (rs.next()) {
        output.append(rs.getString(1));
        output.append("<br>");
      }
    }
    catch (Exception e) {
      output.append("Exception: ");
      output.append(e.getMessage());
      output.append("<br>");
    }
    finally {
      try {
        if (rs != null) {
          rs.close();
        }
        if (stmt != null) {
          stmt.close();
        }
        if (con != null) {
          con.close();
        }
      }
      catch (Exception e) {
        output.append("Exception (during close of connection): ");
        output.append(e.getMessage());
        output.append("<br>");
      }
    }
  %>
```

```
    <%= output.toString() %>
  </body>
</html>
```

AWS OpsWorks Stacks crée et initialise l'`DataSource`objet, le lie à un nom logique et enregistre le nom auprès d'un service de dénomination Java Naming and Directory Interface (JNDI). Le nom logique complet est `java:comp/env/`*`user-assigned-name`*. Vous devez spécifier la partie du nom assignée par l'utilisateur en ajoutant des attributs personnalisés JSON aux attributs de configuration et de déploiement de la pile pour définir l'attribut `['opsworks_java']['datasources']`, comme décrit dans ce qui suit.

Pour déployer une page JSP qui se connecte à une base de données MySQL

1. Créez une pile avec une couche Java App Server, ajoutez des instances 24 h/24 et 7 j/7 à chaque couche, puis démarrez-la.
2. Ajoutez une couche base de données à la pile. Les détails dépendent de la base de données que vous utilisez.

   Pour utiliser une instance MySQL dans cet exemple, ajoutez une couche MySQL à la pile, ajoutez une instance 24 h/24 et 7 j/7 à la couche et démarrez-la.

   Pour utiliser une instance Amazon RDS (MySQL) dans l'exemple, procédez comme suit :

   - Spécifiez un moteur de base de données MySQL pour l'instance.
   - Attribuez les groupes de sécurité AWS- OpsWorks -DB-Master-Server () ***`security_group_id`*** et AWS- OpsWorks -Java-App-Server () à l'instance. *`security_group_id`* AWS OpsWorks Stacks crée ces groupes de sécurité pour vous lorsque vous créez votre première pile dans la région.
   - Créez une base de données nommée `simplejspdb`.
   - Assurez-vous que le nom et le mot de passe de l'utilisateur principal ne contiennent pas le caractère & ou autres caractères susceptibles d'entraîner une erreur Tomcat.

     En particulier, au moment du démarrage Tomcat doit analyser le fichier de contexte d'application web, lequel est un fichier XML incluant le nom d'utilisateur principal et le mot de passe principal. Si l'une ou l'autre des chaînes inclut un caractère &, l'analyseur XML le traite comme une entité XML mal formée et lève une exception d'analyse, ce qui empêche le

démarrage de Tomcat. Pour plus d'informations sur le fichier de contexte d'application web, consultez tomcat::context.

- Ajoutez un pilote MySQL à la couche Java App Server.
- Enregistrez l'instance RDS auprès de votre pile.

Pour plus d'informations sur l'utilisation des instances Amazon RDS avec AWS OpsWorks Stacks, consultez. Couche de service Amazon RDS

3. Copiez l'exemple de code dans un fichier nommé `simplejspdb.jsp`, placez le fichier dans un dossier nommé `simplejspdb` et créez une archive `.zip` du dossier. Les noms sont arbitraires ; vous pouvez utiliser n'importe quels noms de fichier ou de dossier de votre choix. Vous pouvez également utiliser d'autres types d'archives, y compris gzip, bzip2 et tarball. Pour déployer plusieurs pages JSP, incluez-les dans la même archive. Pour plus d'informations sur le déploiement d'applications à partir d'autres types de référentiel, consultez Source de l'application.
4. Téléchargez l'archive dans un compartiment Amazon S3 et rendez le fichier public. Copiez l'URL du fichier pour une utilisation ultérieure. Pour plus d'informations sur la création de compartiments et le chargement des fichiers, consultez Mise en route avec Amazon Simple Storage Service.
5. Ajoutez une application à la pile et spécifiez les paramètres suivants :

   - Nom – `SimpleJSPDB`
   - Type d'application — `Java`
   - Type de source de données — OpsWorks(pour une instance MySQL) ou RDS (pour une instance Amazon RDS).
   - Instance de base de données — L'instance MySQL que vous avez créée précédemment, généralement nommée db-master1 (mysql), ou l'instance Amazon RDS, qui sera nommée ***`DB_instance_name`*** (mysql).
   - Nom de la base de données —`simplejspdb`.
   - Type de référentiel — `Http Archive`
   - URL du référentiel : URL Amazon S3 de votre fichier d'archive.

   Utilisez les valeurs par défaut pour les paramètres restants, puis cliquez sur Ajouter une application pour créer l'application.

6. Ajoutez les attributs suivants du JSON personnalisé aux attributs de configuration de la pile, où simplejspdb correspond au nom court de l'application.

```
{
  "opsworks_java": {
    "datasources": {
      "simplejspdb": "jdbc/mydb"
    }
  }
}
```

   AWS OpsWorks Stacks utilise ce mappage pour générer un fichier de contexte contenant les informations de base de données nécessaires.

   Pour plus d'informations sur l'ajout d'attributs de JSON personnalisé aux attributs de configuration de la pile, consultez Utilisation du JSON personnalisé.

7. Déployez l'application sur l'instance Java App Server.

Vous pouvez désormais utiliser l'URL de l'application pour afficher l'application elle-même. Pour une description de la construction de l'URL, consultez Déploiement d'une application JSP.

L'URL de l'exemple doit ressembler à quelque chose comme `http://192.0.2.0/simplejspdb/simplejspdb.jsp`.

> **Note**
>
> L'attribut `datasources` peut contenir plusieurs attributs. Chaque attribut est nommé avec un nom court d'application et défini avec la partie appropriée, assignée par l'utilisateur, d'un nom logique. Si vous avez plusieurs applications, vous pouvez utiliser des noms logiques distincts, ce qui nécessite un JSON personnalisé tel que celui qui suit.
>
> ```
> {
>   "opsworks_java": {
>     "datasources": {
>       "myjavaapp": "jdbc/myappdb",
>       "simplejsp": "jdbc/myjspdb",
>       ...
>     }
> ```

```
  }
}
```

Node.js App Server AWS OpsWorks Stacks Layer

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour les piles Linux.

La couche Node.js App Server est une couche AWS OpsWorks Stacks qui fournit un modèle pour les instances qui fonctionnent comme des serveurs d'applications Node.js. AWS OpsWorks Stacks installe également Express, de sorte que les instances de la couche prennent en charge à la fois les applications standard et Express.

Installation : Node.js est installé dans `/usr/local/bin/node`.

La page Add Layer (Ajouter une couche) fournit les options de configuration suivantes :

Node.js version (Version de Node.js)

Pour afficher la liste des versions actuellement prises en charge, consultez AWS OpsWorks Systèmes d'exploitation Stacks.

Custom security groups

Ce paramètre apparaît si vous avez choisi de ne pas associer automatiquement un groupe de sécurité AWS OpsWorks Stacks intégré à vos couches. Vous devez spécifier le groupe de sécurité à associer à la couche. Pour de plus amples informations, veuillez consulter Créer une pile.

Elastic Load Balancer

Vous pouvez associer un équilibreur de charge Elastic Load Balancing aux instances de la couche.

> **Important**
>
> Si votre application Node.js utilise le protocole SSL, nous vous recommandons de le désactiver SSLv3 si possible pour corriger les vulnérabilités décrites dans le document CVE-2015-8027. Pour ce faire, vous devez définir Node.js version avec la valeur `0.12.9`.

Déploiement d'applications Node.js

Pour une présentation détaillée de l'implémentation d'une application simple Node.js pour AWS OpsWorks Stacks et de son déploiement sur une pile, consultez Création de votre première pile Node.js. En général, les applications Node.js pour AWS OpsWorks Stacks doivent répondre aux conditions suivantes :

- Le fichier principal doit être nommé `server.js` et résider dans le répertoire racine de l'application déployée.
- Les applications Express doivent inclure un fichier `package.json` dans le répertoire racine de l'application.
- Par défaut, l'application doit écouter sur le port 80 (HTTP) ou le port 443 (HTTPS).

  Il est possible d'écouter sur d'autres ports, mais le groupe de sécurité intégré à la couche Node.js App Server, AWS- OpsWorks -NodeJS-App-Server, autorise le trafic utilisateur entrant uniquement vers les ports 80, 443 et 22 (SSH). Pour autoriser le trafic utilisateur entrant vers d'autres ports, créez un groupe de sécurité avec des règles de trafic entrant appropriées et attribuez-le à la couche Node.js App Server. Ne changez pas les règles de trafic entrant en modifiant le groupe de sécurité intégré. Chaque fois que vous créez une pile, AWS OpsWorks Stacks remplace les groupes de sécurité intégrés par les paramètres standard, de sorte que toutes les modifications que vous apportez seront perdues.

> **Note**
>
> AWS OpsWorks Stacks définit la variable d'environnement PORT sur 80 (par défaut) ou 443 (si vous activez le protocole SSL). Vous pouvez donc utiliser le code suivant pour écouter les demandes.
>
> ```
> app.listen(process.env.PORT);
> ```

Si vous configurez une application Node.js pour prendre en charge le protocole SSL, vous devez spécifier la clé et les certificats. AWS OpsWorks Stacks place les données de chaque instance de serveur d'applications sous forme de fichiers distincts dans le `/srv/www/`*`app_shortname`*`/shared/config` répertoire, comme suit.

- `ssl.crt`— le certificat SSL.
- `ssl.key`— la clé SSL.
- `ssl.ca`— le certificat de chaîne, si vous en avez spécifié un.

Votre application peut obtenir la clé SSL et les certificats à partir de ces fichiers.

couche d' AWS OpsWorks empilements de serveurs d'applications PHP

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour les piles Linux.

La couche PHP App Server est une couche AWS OpsWorks Stacks qui fournit un modèle pour les instances qui fonctionnent comme des serveurs d'applications PHP. La couche PHP App Server est basée sur Apache2 avec mod_php et ne possède aucune option de configuration standard. Les versions PHP et Apache dépendent du système d'exploitation que vous spécifiez pour les instances de la couche.

Système d'exploitation	Version PHP	Version Apache
Amazon Linux 2018.03	5.3	2.2
Amazon Linux 2017.09	5.3	2.2
Amazon Linux 2017.03	5.3	2.2
Amazon Linux 2016.09	5.3	2.2
Amazon Linux 2016.03	5.3	2.2
Amazon Linux 2015.09	5.3	2.2
Amazon Linux 2015.03	5.3	2.2
Amazon Linux 2014.09	5.3	2.2
Ubuntu 14.04 LTS	5.5	2.4

Installation : AWS OpsWorks Stacks utilise le programme d'installation du package de l'instance pour installer Apache2 et mod_php dans ses emplacements par défaut. Pour plus d'informations sur l'installation, consultez Apache.

La page Add Layer (Ajouter une couche) fournit les options de configuration suivantes :

Custom security groups

Ce paramètre apparaît si vous avez choisi de ne pas associer automatiquement un groupe de sécurité AWS OpsWorks Stacks intégré à vos couches. Vous devez spécifier le groupe de sécurité à associer à la couche. Pour de plus amples informations, veuillez consulter Créer une pile.

Elastic Load Balancer

Vous pouvez associer un équilibreur de charge Elastic Load Balancing aux instances de la couche.

Vous pouvez modifier certains paramètres de configuration Apache en utilisant JSON personnalisé ou un fichier d'attributs personnalisé. Pour de plus amples informations, veuillez consulter Remplacement des attributs. Pour obtenir la liste des attributs Apache qui peuvent être remplacés, consultez Attributs apache2.

Pour obtenir un exemple de déploiement d'une application PHP, y compris la connexion de l'application à une base de données principale, consultez Mise en route des piles Linux Chef 11.

Important

Si votre application PHP utilise le protocole SSL, nous vous recommandons de le désactiver SSLv3 si possible pour corriger les vulnérabilités décrites dans CVE-2014-3566. Pour ce faire, vous devez modifier le paramètre `SSLProtocol` dans le fichier `ssl.conf` du serveur Apache. Pour plus d'informations sur la modification de cet attribut, consultez Désactivation SSLv3 pour les serveurs Apache.

Rails App Server AWS OpsWorks Stacks Layer

Important

Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Note

Cette couche est disponible uniquement pour les piles Linux.

La couche Rails App Server est une couche AWS OpsWorks Stacks qui fournit un modèle pour les instances qui fonctionnent comme des serveurs d'applications Rails.

Installation : AWS OpsWorks Stacks utilise le programme d'installation des packages de l'instance pour installer les packages du serveur à leur emplacement par défaut. Pour plus d'informations sur Apache/Passenger l'installation, consultez Phusion Passenger. Pour plus d'informations sur la journalisation, consultez Fichiers journaux. Pour plus d'informations sur Nginx/Unicorn l'installation, consultez Unicorn.

La page Add Layer (Ajouter une couche) fournit les options de configuration suivantes, toutes étant facultatives.

Ruby Version (Version de Ruby)

Version de Ruby qui sera utilisée par vos applications. La valeur par défaut est 2.3.

Vous pouvez aussi spécifier votre version Ruby préférée en remplaçant l'attribut `[:opsworks][:ruby_version]`.

> **Note**
>
> AWS OpsWorks Stacks installe un package Ruby distinct à utiliser par les recettes et l'agent d'instance. Pour de plus amples informations, veuillez consulter Versions de Ruby.

Rails Stack (Pile Rails)

La pile Rails par défaut est Apache2 avec Phusion Passenger. Vous pouvez aussi utiliser Nginx avec Unicorn.

> **Note**
>
> Si vous utilisez Nginx et Unicorn, vous devez ajouter le GEM Unicorn au fichier Gemfile de votre application, comme dans l'exemple suivant :

```
source 'https://rubygems.org'
gem 'rails', '3.2.15'
...
# Use unicorn as the app server
gem 'unicorn'
```

```
...
```

Passenger Version (Version de Passenger)

Si vous avez spécifié Apache2/Passenger, vous devez spécifier la version de Passenger. La valeur par défaut est 5.0.28.

Rubygems Version (Version de Rubygems)

La version de Rubygems par défaut est 2.5.1.

Install and Manage Bundler (Installer et gérer Bundler)

Permet de choisir s'il convient d'installer et de gérer Bundler. La valeur par défaut est Yes (Oui).

Bundler version (Version de Bundler)

La version Bundler par défaut est 1.12.5.

Custom security groups

Ce paramètre apparaît si vous avez choisi de ne pas associer automatiquement un groupe de sécurité AWS OpsWorks Stacks intégré à vos couches. Vous devez spécifier le groupe de sécurité à associer à la couche. Pour de plus amples informations, veuillez consulter Créer une pile.

Elastic Load Balancer

Vous pouvez associer un équilibreur de charge Elastic Load Balancing aux instances de la couche.

Vous pouvez modifier certains paramètres de configuration en utilisant JSON personnalisé ou un fichier d'attributs personnalisé. Pour de plus amples informations, veuillez consulter Remplacement des attributs. Pour obtenir la liste des attributs Apache, Nginx, Phusion Passenger et Unicorn qui peuvent être remplacés, consultez Attributs des livres de recettes intégrés.

Important

Si votre application Ruby on Rails utilise le protocole SSL, nous vous recommandons de le désactiver SSLv3 si possible pour corriger les vulnérabilités décrites dans CVE-2014-3566. Pour de plus amples informations, veuillez consulter Désactivation SSLv3 pour les serveurs Rails.

Rubriques

- Désactivation SSLv3 pour les serveurs Rails
- Connexion à une base de données
- Déploiement des applications Ruby on Rails

Désactivation SSLv3 pour les serveurs Rails

Pour le désactiver SSLv3 pour les serveurs Rails, mettez à jour le paramètre de version Ruby de la couche vers 2.1 ou supérieur, ce qui installe Ruby 2.1.4 ou supérieur comme version utilisée par les applications.

- Mettez à jour le paramètre Ruby Version (Version de Ruby) avec la valeur 2.1 ou supérieure.
- Mettez à jour le fichier de configuration de votre pile Rails comme suit.

Apache avec Phusion Passenger

Mettez à jour le paramètre `SSLProtocol` du fichier `ssl.conf` du serveur Apache, comme décrit dans Désactivation SSLv3 pour les serveurs Apache.

Nginx avec Unicorn

Ajoutez une directive `ssl_protocols` explicite au fichier `nginx.conf` du serveur Nginx. Pour le désactiver SSLv3, remplacez le fichier modèle du livre de recettes nginx intégré, que les recettes `nginx.conf.erb` de configuration de la couche Rails App Server utilisent pour `nginx.conf` créer, et ajoutez la directive suivante :

```
ssl_protocols TLSv1.2;
```

Pour plus d'informations sur la configuration de `nginx.conf`, consultez Configuration des serveurs HTTPS. Pour plus d'informations sur le remplacement d'un modèle intégré, consultez Utilisation de modèles personnalisés.

Connexion à une base de données

Lorsque vous déployez une application, AWS OpsWorks Stacks crée un nouveau `database.yml` fichier en utilisant les informations provenant des `deploy`attributs de l'application. Si vous attachez une instance MySQL ou Amazon RDS à l'application, AWS OpsWorks Stacks ajoute les informations

de connexion aux `deploy` attributs, afin qu'elles contiennent `database.yml` automatiquement les données de connexion correctes.

Si aucune base de données n'est attachée à une application, par défaut, AWS OpsWorks Stacks n'ajoute aucune information de connexion aux `deploy` attributs et ne crée `database.yml` pas. Si vous souhaitez utiliser une autre base de données, vous pouvez utiliser JSON personnalisé pour ajouter les attributs de base de données aux attributs `deploy` de l'application avec les informations de connexion. Les attributs se trouvent tous en dessous`["deploy"]["`*`appshortname`*`"]["database"]`, où se *`appshortname`* trouve le nom abrégé de l'application, que AWS OpsWorks Stacks génère à partir du nom de l'application. Les valeurs que vous spécifiez dans JSON personnalisé remplacent les paramètres par défaut. Pour de plus amples informations, veuillez consulter Ajout d'applications.

AWS OpsWorks Stacks intègre les valeurs `[:...][:database]`d'attribut suivantes dans. `database.yml` Les attributs requis dépendent de la base de données en question, mais vous devez disposer d'un `host` attribut, sinon AWS OpsWorks Stacks n'en créera `database.yml` pas.

- `[:adapter] (String)`— L'adaptateur de base de données, tel que`mysql`.
- `[:database]`(String) — Nom de la base de données.
- `[:encoding]`(String) — Le codage, qui est généralement défini sur`utf8`.
- `[:host]`(String) — L'URL de l'hôte, telle que`railsexample.cdlqlk5uwd0k.us-west-2.rds.amazonaws.com`.
- `[:reconnect]`(Boolean) — Indique si l'application doit se reconnecter si la connexion n'existe plus.
- `[:password]`(String) — Le mot de passe de la base de données.
- `[:port]` (Numéro). — Le numéro de port de la base de données. Utilisez cet attribut pour remplacer le numéro de port par défaut, qui est défini par l'adaptateur.
- `[:username]`(String) — Nom d'utilisateur de la base de données.

L'exemple suivant montre JSON personnalisé pour une application dont le nom court est myapp.

```
{
  "deploy" : {
    "myapp" : {
      "database" : {
```

```
        "adapter" : "adapter",
        "database" : "databasename",
        "host" : "host",
        "password" : "password",
        "port" : portnumber
        "reconnect" : true/false,
        "username" : "username"
      }
    }
  }
}
```

Pour plus d'informations sur la spécification du JSON personnalisé, consultez Utilisation du JSON personnalisé. Pour voir le modèle utilisé pour créer `database.yml` (`database.yml.erb`), accédez au référentiel du livre de recettes intégré.

Déploiement des applications Ruby on Rails

Vous pouvez déployer les applications Ruby on Rails à partir de n'importe lequel des référentiels pris en charge. Ce qui suit montre comment déployer un exemple d'application Ruby on Rails sur un serveur exécutant une pile Apache/Passenger Rails. L'exemple de code est stocké dans un GitHub référentiel public, mais la procédure de base est la même pour les autres référentiels pris en charge. Pour plus d'informations sur la façon de créer et de déployer des applications, consultez Applications. Pour voir le code de l'exemple, qui inclut de nombreux commentaires, accédez à https://github.com/awslabs/opsworks-demo-rails-photo-share-app.

Pour déployer une application Ruby on Rails à partir d'un GitHub référentiel

1. Créez une pile avec une couche Rails App Server utilisée Apache/Passenger comme pile Rails, ajoutez une instance 24 heures sur 24, 7 jours sur 7 à la couche et démarrez-la.
2. Une fois que l'instance est en ligne, ajoutez une application à la pile et spécifiez les paramètres suivants :

   - Nom — Le nom que vous préférez ; l'exemple utilise`PhotoPoll`.

     AWS OpsWorks Stacks utilise ce nom à des fins d'affichage et génère un nom abrégé pour un usage interne et pour identifier l'application dans la configuration de la pile et les attributs de déploiement. Par exemple, le nom PhotoPoll abrégé est photopoll.
   - Type d'application : Ruby on Rails.
   - Environnement Rails — Les environnements disponibles sont déterminés par l'application.

L'exemple d'application en possède trois : **`development`**, **`test`** et **`production`**. Pour cet exemple, définissez l'environnement avec la valeur **`development`**. Consultez l'exemple de code pour une description de chaque environnement.

- Type de référentiel : tous les types de référentiels pris en charge. Spécifiez `Git` pour cet exemple.
- URL du référentiel : référentiel à partir duquel le code doit être déployé.

  Pour cet exemple, définissez l'URL avec la valeur **`git://github.com/awslabs/opsworks-demo-rails-photo-share-app`**.

Utilisez les valeurs par défaut pour les paramètres restants, puis cliquez sur Ajouter une application pour créer l'application.

3. Déployez l'application sur l'instance Rails App Server.
4. Lorsque le déploiement est terminé, accédez à la page Instances et cliquez sur l'adresse IP publique de l'instance Rails App Server. Vous devez voir ce qui suit :



Couche Static Web Server AWS OpsWorks Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour les piles Linux.

La couche Static Web Server est une couche AWS OpsWorks Stacks qui fournit un modèle pour les instances destinées à diffuser des pages HTML statiques, ce qui peut inclure des scripts côté client. Cette couche s'appuie sur Nginx.

Installation : Nginx est installé dans `/usr/sbin/nginx`.

La page Add Layer (Ajouter une couche) fournit les options de configuration suivantes :

Custom security groups

Ce paramètre apparaît si vous avez choisi de ne pas associer automatiquement un groupe de sécurité AWS OpsWorks Stacks intégré à vos couches. Vous devez spécifier le groupe de sécurité à associer à la couche. Pour de plus amples informations, veuillez consulter Créer une pile.

Elastic Load Balancer

Vous pouvez associer un équilibreur de charge Elastic Load Balancing aux instances de la couche.

Vous pouvez modifier certains paramètres de configuration Nginx en utilisant JSON personnalisé ou un fichier d'attributs personnalisé. Pour de plus amples informations, veuillez consulter Remplacement des attributs. Pour obtenir la liste des attributs Apache qui peuvent être remplacés, consultez Attributs nginx.

Important

Si votre application Web utilise le protocole SSL, nous vous recommandons de le désactiver SSLv3 si possible pour corriger les vulnérabilités décrites dans le document CVE-2014-3566. Pour le désactiver SSLv3, vous devez modifier le fichier du serveur Nginx. `nginx.conf` Pour ce faire, remplacez le fichier modèle du livre de recettes nginx intégré, que les recettes `nginx.conf.erb` de configuration de la couche Rails App Server utilisent pour `nginx.conf` créer, et ajoutez la directive suivante :

```
ssl_protocols TLSv1.2;
```

Pour plus d'informations sur la configuration de `nginx.conf`, consultez Configuration des serveurs HTTPS. Pour plus d'informations sur le remplacement d'un modèle intégré, consultez Utilisation de modèles personnalisés.

Référence de couche de cluster ECS

Note

Cette couche est disponible uniquement pour les piles Linux.

Une couche de cluster ECS représente un cluster Amazon Elastic Container Service (Amazon ECS) et simplifie la gestion du cluster.

Nom court : ecs-cluster

Compatibilité : une couche de service Amazon ECS est compatible uniquement avec les couches personnalisées

Ports ouverts : le cluster ECS autorise l'accès public au port 22 (SSH)

Autoassign Elastic IP addresses : Désactivé par défaut

Default EBS volume : Non

Groupe de sécurité par défaut : AWS-OpsWorks-ECS-Cluster

Configuration : pour configurer une couche de cluster ECS, vous devez spécifier les éléments suivants :

- S'il convient d'attribuer des adresses IP publiques ou des adresses IP Elastic aux instances de conteneur
- Profil d'instance pour les instances de conteneur

Recettes Setup :

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- opsworks_ecs::setup

Recettes Configure :

- opsworks_ganglia::configure-client
- ssh_users
- mysql::client
- agent_version
- opsworks_ecs::configure

Recettes Deploy :

- deploy::default
- opsworks_ecs::deploy

Recettes Undeploy :

- opsworks_ecs::undeploy

Recettes Shutdown :

- opsworks_shutdown::default
- opsworks_ecs::shutdown

Installation:

- AWS OpsWorks Stacks utilise le programme d'installation du package de l'instance pour installer Docker dans ses emplacements par défaut
- Le journal Chef de l'événement d'installation indique si l'agent Amazon ECS a été correctement installé. Dans le cas contraire, les journaux fournis par AWS OpsWorks Stacks n'incluent pas les informations du journal d'erreurs Amazon ECS. Pour plus d'informations sur la façon de gérer les erreurs Amazon ECS, consultez la section Dépannage d'Amazon ECS.

Référence de couche personnalisée

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Si les couches standard ne répondent pas à vos besoins, vous pouvez créer une couche personnalisée. Une pile peut avoir plusieurs couches personnalisées. Par défaut, la couche personnalisée exécute un ensemble limité de recettes standard qui prennent en charge les fonctionnalités de base. Vous implémentez ensuite les principales fonctionnalités de la couche en mettant en œuvre un ensemble de recettes Chef personnalisées pour chacun des événements de cycle de vie approprié afin d'installer et de configurer les logiciels de la couche, et ainsi de suite. Les recettes personnalisées suivent les recettes standard de AWS OpsWorks Stacks pour chaque événement.

Nom court : défini par l'utilisateur ; chaque couche personnalisée d'une pile doit avoir un nom court différent

Ports ouverts : par défaut, une couche serveur personnalisée ouvre l'accès public aux ports 22 (SSH), 80 (HTTP), 443 (HTTPS) et tous les ports des couches serveurs d'applications Rails et PHP de la couche

Autoassign Elastic IP Addresses : Désactivé par défaut

Default EBS volume : Non

Groupe de sécurité par défaut : AWS-OpsWorks-Custom-Server

Compatibilité : les couches personnalisées sont compatibles avec les couches suivantes : personnalisée, db-master, lb, memcached, monitoring-master, nodejs-app, php-app, rails-app et web.

Configuration : Pour configurer une couche personnalisée, vous devez spécifier les valeurs suivantes :

- Nom de la couche
- Nom court de la couche, qui identifie la couche dans les recettes Chef et ne doit utiliser que les lettres a-z et les chiffres

Pour les piles Linux, la couche personnalisée utilise les recettes suivantes.

Recettes Setup :

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client

Recettes Configure :

- opsworks_ganglia::configure-client
- ssh_users

- agent_version

Recettes Deploy :

- deploy::default

Recettes Shutdown :

- opsworks_shutdown::default

Autres couches de référence

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks prend également en charge les couches suivantes.

Rubriques

- Référence de la couche Ganglia
- Référence de couche Memcached

Référence de la couche Ganglia

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour les piles Linux.

Une couche Ganglia prend en charge Ganglia, un système de surveillance distribué qui gère le stockage et la visualisation des métriques d'instance. Il est conçu pour travailler avec les topologies d'instances hiérarchiques, ce qui le rend particulièrement utile pour les groupes d'instances. Ganglia possède deux composants de base :

- Un client à faible charge, qui est installé sur chaque instance de la pile et envoie les métriques au maître.
- Un master, qui collecte les métriques des clients et les stocke sur un volume Amazon EBS. Il affiche également les métriques sur une page web.

AWS OpsWorks Stacks dispose d'un agent de surveillance Ganglia sur chaque instance qu'il gère. Lorsque vous ajoutez une couche Ganglia à votre stack et que vous la démarrez, les agents Ganglia de chaque instance transmettent des métriques à l'instance Ganglia. Pour utiliser Ganglia, ajoutez une couche Ganglia avec une instance à la pile. Vous accédez aux données en vous connectant au serveur Ganglia à l'adresse IP du maître. Vous pouvez fournir des définitions de métriques supplémentaires en écrivant des recettes Chef.

Nom court : monitoring-master

Compatibilité : Une couche Ganglia est compatible avec les couches suivantes : custom, db-master, memcached, php-app, rails-app.

Ports ouverts : l'équilibreur de charge autorise l'accès public aux ports 22 (SSH), 80 (HTTP) et 443 (HTTPS).

Autoassign Elastic IP addresses : Désactivé par défaut

Default EBS volume (Volume EBS par défaut) : Oui, dans `/vol/ganglia`

Groupe de sécurité par défaut : AWS-OpsWorks-Monitoring-Master -Server

Configuration : Pour configurer une couche Ganglia, vous devez spécifier les éléments suivants :

- L'URI qui fournit un accès aux graphiques de supervision. La valeur par défaut est http ://*DNSName*/ganglia, où *DNSName* est le nom DNS de l'instance Ganglia.

- Un nom d'utilisateur et un mot de passe qui contrôlent l'accès aux statistiques de surveillance.

Recettes Setup :

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- opsworks_ganglia::server

Recettes Configure :

- opsworks_ganglia::configure-client
- ssh_users
- agent_version
- opsworks_ganglia::configure-server

Recettes Deploy :

- deploy::default
- opsworks_ganglia::configure-server
- opsworks_ganglia::deploy

Recettes Shutdown :

- opsworks_shutdown::default
- apache2::stop

Installation:

- Le client Ganglia est installé sous : `/etc/ganglia`.

- Le serveur web frontal Ganglia est installé sous : `/usr/share/ganglia-webfrontend`.
- Le logtailer Ganglia est installé sous : `/usr/share/ganglia-logtailer`.

Référence de couche Memcached

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour les piles Linux.

Memcached est un système distribué de mise en mémoire cache pour les données arbitraires. Il accélère la vitesse des sites web en mettant en cache les chaînes et les objets comme clés et valeurs dans la mémoire pour réduire le nombre de fois où une source externe de données doit être lue.

Pour utiliser Memcached dans une pile, créez une couche Memcached et ajoutez une ou plusieurs instances, qui fonctionnent comme des serveurs Memcached. Les instances installent automatiquement Memcached et les autres instances de la pile peuvent accéder aux serveurs Memcached et les utiliser. Si vous utilisez une couche Rails App Server, AWS OpsWorks Stacks place automatiquement un fichier `memcached.yml` de configuration dans le répertoire de configuration de chaque instance de la couche. Vous pouvez obtenir le serveur Memcached et le numéro de port à partir de ce fichier.

Nom court : memcached

Compatibilité : une couche Memcached est compatible avec les couches suivantes : custom, db-master, lb, monitoring-master, nodejs-app, php-app, rails-app et web.

Ports ouverts : une couche Memcached permet l'accès public au port 22 (SSH) et à tous les ports depuis les serveurs Web, les serveurs personnalisés et les serveurs d'applications Rails, PHP et Node.js de la pile.

Autoassign Elastic IP addresses : Désactivé par défaut

Default EBS volume : Non

Groupe de sécurité par défaut : AWS-OpsWorks-Memcached-Server

Pour configurer une couche Memcached, vous devez spécifier la taille du cache, en Mo.

Recettes Setup :

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql::client
- dependencies
- ebs
- opsworks_ganglia::client
- memcached

Recettes Configure :

- opsworks_ganglia::configure-client
- ssh_users
- agent_version

Recettes Deploy :

- deploy::default

Recettes Shutdown :

- opsworks_shutdown::default
- memcached::stop

Installation:

- AWS OpsWorks Stacks utilise le programme d'installation du package de l'instance pour installer Memcached et ses fichiers journaux dans leurs emplacements par défaut.

Autres couches

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Ces couches sont disponibles uniquement pour les piles Linux de Chef 11 et des versions antérieures.

AWS OpsWorks Stacks prend également en charge les couches Ganglia et Memcached.

Rubriques

- Couche ganglionnaire
- Memcached

Couche ganglionnaire

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour Chef 11 et les piles antérieures basées sur Linux.

AWS OpsWorks Stacks envoie toutes les statistiques de votre instance et de votre volume à Amazon CloudWatch, ce qui vous permet de visualiser facilement des graphiques et de définir des alarmes pour vous aider à résoudre les problèmes et à prendre des mesures automatisées en fonction de l'état de vos ressources. Vous pouvez également utiliser la couche Ganglia AWS OpsWorks Stacks pour des options supplémentaires de surveillance des applications, telles que le stockage des métriques que vous avez choisies.

La couche Ganglia est un modèle pour une instance qui surveille votre stack à l'aide de la surveillance distribuée de Ganglia. Une pile ne possède généralement qu'une seule instance de Ganglia. La couche Ganglia inclut les paramètres de configuration facultatifs suivants :

URL Ganglia

Le chemin de l'URL de la statistique. L'URL complète est http ://*DNSNameURLPath*, où *DNSName* est le nom DNS de l'instance associée. La *URLPath* valeur par défaut est « /ganglia », ce qui correspond à quelque chose comme : http://ec2-54-245-151-7.us-west-2.compute.amazonaws.com/ganglia.

Nom d'utilisateur Ganglia

Un nom d'utilisateur pour la page web de statistiques. Vous devez fournir le nom d'utilisateur lorsque vous consultez la page. La valeur par défaut est « opsworks ».

Mot de passe Ganglia

Un mot de passe qui contrôle l'accès à la page web des statistiques. Vous devez fournir le mot de passe lorsque vous consultez la page. La valeur par défaut est une chaîne générée de façon aléatoire.

> **Note**
>
> Enregistrez le mot de passe pour une utilisation ultérieure ; AWS OpsWorks Stacks ne vous permet pas de voir le mot de passe après avoir créé la couche. Cependant, vous

pouvez mettre à jour le mot de passe en accédant à la page de modification de la couche et en cliquant sur Mettre à jour le mot de passe.

Custom security groups

Ce paramètre apparaît si vous avez choisi de ne pas associer automatiquement un groupe de sécurité AWS OpsWorks Stacks intégré à vos couches. Vous devez spécifier le groupe de sécurité à associer à la couche. Pour de plus amples informations, veuillez consulter Créer une pile.

Elastic Load Balancer

Vous pouvez associer un équilibreur de charge Elastic Load Balancing aux instances de la couche.

Important

Si votre stack inclut une couche Ganglia, nous vous recommandons de la désactiver SSLv3 si possible pour que cette couche corrige les vulnérabilités décrites dans CVE-2014-3566. Pour ce faire, vous devez remplacer le modèle `ssl.conf.erb` du serveur Apache pour modifier le paramètre `SSLProtocol`. Pour en savoir plus, consultez Désactivation SSLv3 pour les serveurs Apache.

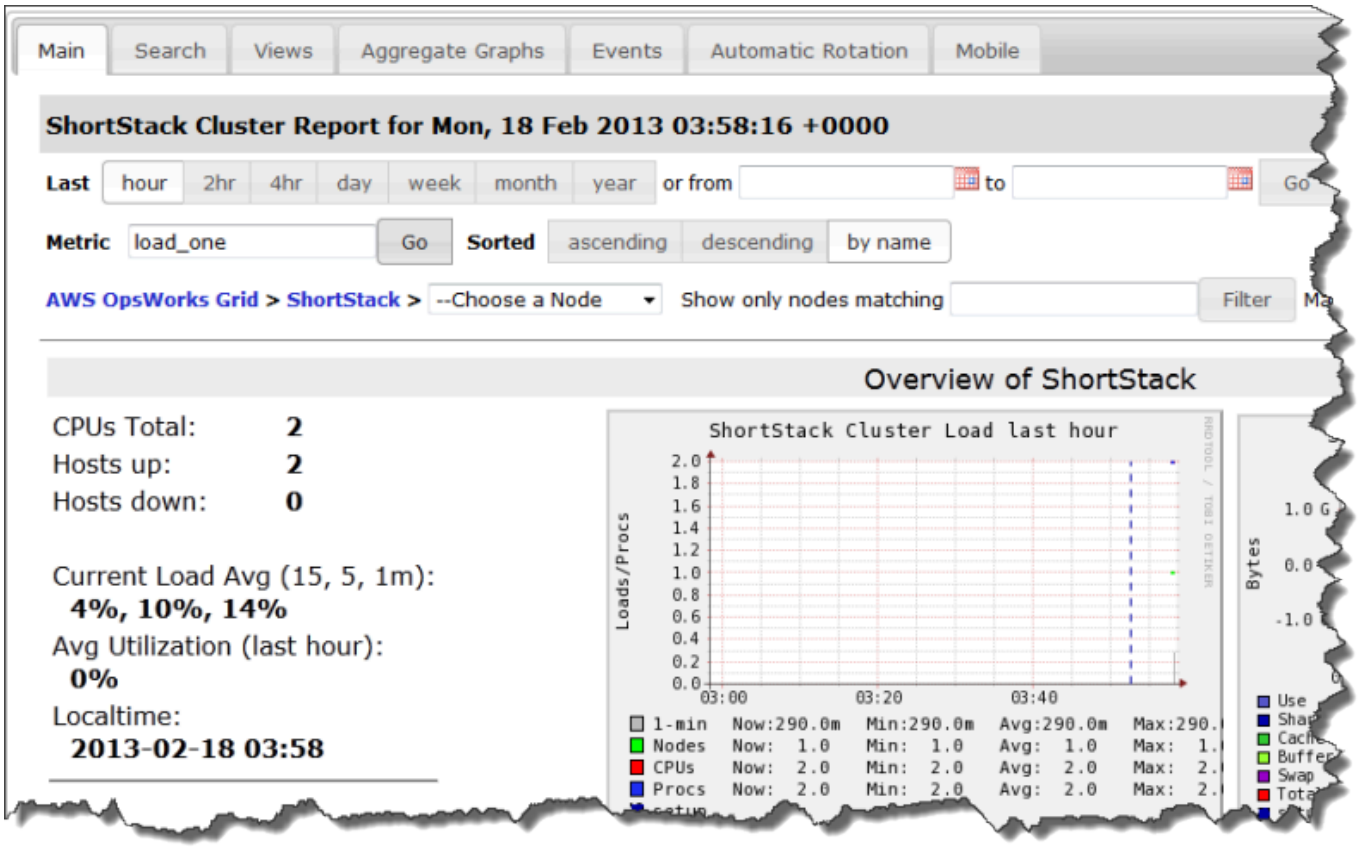
Afficher les statistiques sur les ganglions

AWS OpsWorks Les recettes Stacks installent un client Ganglia à faible coût sur chaque instance. Si votre stack inclut une couche Ganglia, le client Ganglia commence automatiquement à envoyer des rapports au Ganglia dès que l'instance est en ligne. The Ganglia utilise les données du client pour calculer diverses statistiques et affiche les résultats sous forme graphique sur sa page Web de statistiques.

Pour afficher les statistiques Ganglia

1. Sur la page Couches, cliquez sur Ganglia pour ouvrir la page de détails de la couche.
2. Dans le volet de navigation, cliquez sur Instances. Sous Ganglia, cliquez sur le nom de l'instance.

3. Copiez le nom DNS public de l'instance.
4. Utilisez le nom DNS pour créer l'URL des statistiques, qui ressemblera à : http://ec2-54-245-151-7.us-west-2.compute.amazonaws.com/ganglia.
5. Collez l'URL complète dans votre navigateur, accédez à la page et entrez le nom d'utilisateur et le mot de passe Ganglia pour afficher la page. Un exemple suit.



Memcached

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette couche est disponible uniquement pour les piles Linux de Chef 11 et des versions antérieures.

Une couche Memcached est une couche AWS OpsWorks Stacks qui fournit un modèle pour les instances qui fonctionnent comme des serveurs Memcached, c'est-à-dire un système de mise en cache de mémoire distribué pour des données arbitraires. La couche Memcached inclut les paramètres de configuration suivants.

Mémoire allouée (Mo)

(Facultatif) La quantité de mémoire cache (en Mo) pour chacune des instances de la couche. La valeur par défaut est 512 Mo.

Custom security groups

Ce paramètre apparaît si vous avez choisi de ne pas associer automatiquement un groupe de sécurité AWS OpsWorks Stacks intégré à vos couches. Vous devez spécifier le groupe de sécurité à associer à la couche. Pour de plus amples informations, veuillez consulter Créer une pile.

Composants des livres de recettes

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Un livre de recettes inclut généralement les éléments de base suivants :

- Les fichiers Attribut contiennent un ensemble d'attributs qui représentent les valeurs qu'utilisent les recettes et les modèles.

- Les fichiers Modèle correspondent aux modèles que les recettes utilisent pour créer d'autres fichiers, tels que les fichiers de configuration.

  Les fichiers modèles vous permettent généralement de modifier le fichier de configuration en remplaçant les attributs, ce qui peut être fait sans toucher au livre de recettes, au lieu de réécrire un fichier de configuration. La pratique standard est que chaque fois que vous souhaitez modifier, même légèrement, un fichier de configuration sur une instance, vous devez utiliser un fichier modèle.
- Les fichiers Recette sont des applications Ruby qui définissent tout ce qui est nécessaire pour configurer un système, y compris la création et la configuration de dossiers, l'installation et la configuration de packages, le démarrage de services, et ainsi de suite.

Les livres de recettes n'ont pas nécessairement les trois composants. L'approche la plus simple de la personnalisation nécessite uniquement des fichiers attribut ou modèle. En outre, les livres de recettes peuvent le cas échéant inclure d'autres types de fichiers, tels que les définitions ou les spécifications.

Cette section décrit les trois composants standard des livres de recettes. Pour plus d'informations, notamment sur la façon d'implémenter les recettes, consultez Opscode.

Rubriques

- Attributs
- Modèles
- Recettes

Attributs

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les recettes et les modèles dépendent de différents valeurs, telles que les paramètres de configuration. Plutôt que de coder ces valeurs en dur directement dans les recettes ou les modèles,

vous pouvez créer un fichier attribut avec un attribut représentant chaque valeur. Vous utilisez ensuite les attributs dans vos recettes ou modèles, au lieu des valeurs explicites. L'avantage de l'utilisation des attributs est que vous pouvez remplacer leurs valeurs sans toucher le livre de recettes. Pour cette raison, vous devez toujours utiliser des attributs pour définir les types de valeurs suivants :

- Valeurs susceptibles de varier de pile en pile ou avec le temps, comme les noms d'utilisateur.

  Si vous codez ces valeurs en dur, vous devez modifier le modèle ou la recette chaque fois que vous avez besoin de changer une valeur. En utilisant des attributs pour définir ces valeurs, vous pouvez utiliser les mêmes livres de recettes pour chaque pile et juste remplacer les attributs appropriés.
- Valeurs sensibles, telles que mots de passe ou clés secrètes.

  Le placement de valeurs sensibles explicites dans votre livre de recettes peut accroître le risque d'exposition. Au lieu de cela, définissez les attributs avec des valeurs fictives et remplacez-les pour définir les valeurs réelles. Le meilleur moyen de remplacer ces attributs est un JSON personnalisé. Pour de plus amples informations, veuillez consulter Utilisation du JSON personnalisé.

Pour plus d'informations sur les attributs et leur remplacement, consultez Remplacement des attributs.

L'exemple suivant constitue une partie d'un exemple de fichier attribut.

```
...
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 définit les attributs en utilisant la syntaxe suivante :

```
node.type["attribute"]["subattribute"]["..."]=value
```

Vous pouvez utiliser également le signe deux-points (:), comme suit :

```
node.type[:attribute][:subattribute][:...]=value
```

Une définition d'attribut comprend les éléments suivants :

`node.`

Le préfixe `node.` est facultatif et généralement omis, comme illustré dans l'exemple.

`type`

Le type détermine si l'attribut peut être remplacé. AWS OpsWorks Les attributs Stacks utilisent généralement l'un des types suivants :

- `default` est le type le plus couramment utilisé, car il permet que l'attribut soit remplacé.
- `normal`définit un attribut qui remplace l'une des valeurs d'attribut AWS OpsWorks Stacks standard.

> **Note**
>
> Chef prend en charge des types supplémentaires, qui ne sont pas nécessaires pour AWS OpsWorks Stacks mais qui peuvent être utiles pour votre projet. Pour plus d'informations, consultez À propos des attributs.

`attribute name`

Le nom d'attribut utilise la syntaxe de nœud Chef standard, `[:`*`attribute`*`][:`*`subattribute`*`][...]`. Vous pouvez utiliser n'importe quel nom de votre choix pour vos attributs. Cependant, comme expliqué dans Remplacement des attributs, les attributs de livres de recettes personnalisés sont fusionnés dans l'objet de nœud de l'instance, ainsi que les attributs de configuration et de déploiement de la pile, et l'outil Ohai de Chef. Les noms de configuration généralement utilisés comme port ou utilisateur peuvent apparaître dans un grand nombre de livres de recettes.

Pour éviter les conflits de nom, la convention consiste à créer des noms d'attribut qualifiés avec au moins deux éléments, comme illustré dans l'exemple. Le premier élément doit être unique et est généralement basé sur un nom de produit, comme Apache. Il est suivi d'un ou de plusieurs sous-attributs qui identifient la valeur particulière, comme `[:user]` ou `[:port]`. Vous pouvez utiliser autant de sous-attributs qu'approprié pour votre projet.

`value`

Un attribut peut être défini avec les types de valeurs suivants :

- Une chaîne, telle que `default[:apache][:keepalive] = 'Off'`.
- Un nombre (sans guillemets), tel que `default[:apache][:timeout] = 120`.
- Une valeur booléenne, qui peut être `true` ou `false` (sans guillemets).
- Une liste de valeurs, telle que `default[:apache][:listen_ports] = [ '80','443' ]`

Comme le fichier attribut est une application Ruby, vous pouvez également utiliser les opérateurs logiques et syntaxiques de nœud pour attribuer des valeurs basées sur d'autres attributs. Pour plus d'informations sur la façon de définir les attributs, consultez À propos des attributs. Pour des exemples de fichiers d'attributs fonctionnels, consultez les livres de recettes intégrés à AWS OpsWorks Stacks sur opsworks-cookbooks. https://github.com/aws/

Modèles

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous configurez un grand nombre de packages en créant un fichier de configuration et en le plaçant dans le répertoire approprié. Vous pouvez inclure un fichier de configuration dans votre livre de recettes et le copier dans le répertoire approprié, mais une approche plus souple consiste à ce que vos recettes créent le fichier de configuration à partir d'un modèle. L'un des avantages d'un modèle est que vous pouvez utiliser des attributs pour définir les valeurs du modèle. Cela vous permet, par

exemple, de modifier un fichier de configuration sans toucher le livre de recettes en utilisant un JSON personnalisé pour remplacer les valeurs d'attribut appropriées.

Un modèle possède essentiellement le même contenu et la même structure que le fichier associé. Voici un exemple de fichier : `httpd.conf`.

```
ServerRoot "<%= node[:apache][:dir] %>"
<% if node[:platform] == "debian" || node[:platform] == "ubuntu" -%>
  LockFile /var/lock/apache2/accept.lock
<% else -%>
   LockFile logs/accept.lock
<% end -%>
PidFile <%= node[:apache][:pid_file] %>
Timeout <%= node[:apache][:timeout] %>
KeepAlive <%= node[:apache][:keepalive] %>
MaxKeepAliveRequests <%= node[:apache][:keepaliverequests] %>
KeepAliveTimeout <%= node[:apache][:keepalivetimeout] %>
<IfModule mpm_prefork_module>
    StartServers          <%= node[:apache][:prefork][:startservers] %>
    MinSpareServers       <%= node[:apache][:prefork][:minspareservers] %>
    MaxSpareServers       <%= node[:apache][:prefork][:maxspareservers] %>
    ServerLimit           <%= node[:apache][:prefork][:serverlimit] %>
    MaxClients            <%= node[:apache][:prefork][:maxclients] %>
    MaxRequestsPerChild   <%= node[:apache][:prefork][:maxrequestsperchild] %>
</IfModule>
...
```

L'exemple suivant est le fichier `httpd.conf` qui a été généré pour une instance Ubuntu :

```
ServerRoot "/etc/httpd"
LockFile logs/accept.lock
PidFile /var/run/httpd/httpd.pid
Timeout 120
KeepAlive Off
MaxKeepAliveRequests 100
KeepAliveTimeout 3
<IfModule mpm_prefork_module>
    StartServers          16
    MinSpareServers       16
    MaxSpareServers       32
    ServerLimit           400
```

```
    MaxClients            400
    MaxRequestsPerChild   10000
</IfModule>
...
```

Une grande partie du texte du modèle est simplement copiée à partir du modèle dans le fichier `httpd.conf`. Cependant, le contenu `<%= ... %>` est géré comme suit :

- Chef remplace `<%= node[:attribute][:sub_attribute][:...]%>` par la valeur de l'attribut.

  Par exemple, `StartServers <%= node[:apache][:prefork][:startservers] %>` devient `StartServers 16` dans le fichier `httpd.conf`.
- Vous pouvez utiliser `<%if-%>, <%else-%>, and <%end-%>` de manière conditionnelle pour sélectionner une valeur.

  L'exemple définit un chemin d'accès différent pour `accept.lock` en fonction de la plateforme.

> **Note**
>
> Vous n'êtes pas limité aux attributs des fichiers d'attributs de votre livre de recettes. Vous pouvez utiliser un attribut dans l'objet nœud de l'instance. Par exemple, généré par un outil Chef appelé Ohai et également intégré à l'objet nœud. Pour plus d'informations sur les attributs, consultez Remplacement des attributs.

Pour plus d'informations sur les modèles, y compris sur la façon d'intégrer le code Ruby, consultez À propos des modèles.

Recettes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les recettes sont des applications Ruby qui définissent la configuration d'un système. Elles installent les packages, créent les fichiers de configuration à partir des modèles, exécutent les commandes shell, créent des fichiers et des répertoires, etc. Généralement, AWS OpsWorks Stacks exécute les recettes automatiquement lorsqu'un événement du cycle de vie se produit sur l'instance, mais vous pouvez également les exécuter explicitement à tout moment à l'aide de la commande Execute Recipes stack. Pour plus d'informations, consultez À propos des recettes.

Généralement, une recette se compose d'une série de ressources, chacune représentant l'état souhaité d'un aspect du système. Chaque ressource comprend un ensemble d'attributs qui définissent l'état souhaité et spécifient l'action à entreprendre. Chef associe chaque ressource à un fournisseur approprié qui exécute l'action. Pour plus d'informations, consultez Référence des ressources et fournisseurs.

Une ressource `package` vous permet de gérer les packages logiciels sur les instances Linux. L'exemple suivant installe le package Apache.

```
...
package 'apache2' do
  case node[:platform]
  when 'centos','redhat','fedora','amazon'
    package_name 'httpd'
  when 'debian','ubuntu'
    package_name 'apache2'
  end
  action :install
end
...
```

Chef utilise le fournisseur de packages approprié pour la plateforme. Les attributs de ressource se voient souvent attribuer juste une valeur, mais vous pouvez utiliser les opérations logiques Ruby pour effectuer des assignations conditionnelles. L'exemple utilise un opérateur `case`, qui utilise `node[:platform]` pour identifier le système d'exploitation de l'instance et définit l'attribut `package_name` en conséquence. Vous pouvez insérer des attributs dans une recette en utilisant la syntaxe de nœud Chef standard et Chef remplace l'attribut par la valeur associée. Vous pouvez utiliser n'importe quel attribut de l'objet nœud, pas seulement les attributs de votre livre de recettes.

Après avoir déterminé le nom du package approprié, le segment de code se termine par une action `install`, qui installe le package. D'autres actions pour cette ressource incluent `upgrade` et `remove`. Pour plus d'informations, consultez package.

Il est souvent utile de décomposer les tâches complexes d'installation et de configuration en une ou plusieurs sous-tâches, chacune implémentée comme une recette distincte, et que votre recette principale les exécute au moment approprié. L'exemple suivant illustre la ligne de code qui suit l'exemple précédent :

```
include_recipe 'apache2::service'
```

Pour qu'une recette exécute une recette enfant, utilisez le mot clé `include_recipe`, suivi du nom de la recette. Les recettes sont identifiées à l'aide de la syntaxe Chef *`CookbookName`*`::`*`RecipeName`* standard, où *`RecipeName`* omet l'extension `.rb`.

> **Note**
>
> Une instruction `include_recipe` exécute effectivement la recette à ce stade de la recette principale. Cependant, ce qui se passe réellement est que Chef remplace chaque instruction `include_recipe` par le code de la recette spécifiée avant d'exécuter la recette principale.

Une ressource `directory` représente un répertoire tel que celui qui doit contenir les fichiers d'un package. La ressource `default.rb` suivante crée un répertoire des journaux Linux.

```
directory node[:apache][:log_dir] do
    mode 0755
    action :create
end
```

Le répertoire des journaux est défini dans l'un des fichiers d'attributs du livre de recettes. La ressource spécifie le mode du répertoire comme 0755 et utilise une action `create` pour créer le répertoire. Pour plus d'informations, consultez directory. Vous pouvez également utiliser cette ressource avec les instances Windows.

La ressource `execute` représente les commandes, telles que les commandes shell ou les scripts. L'exemple suivant génère des fichiers module.load.

```
execute 'generate-module-list' do
  if node[:kernel][:machine] == 'x86_64'
```

```
    libdir = 'lib64'
  else
    libdir = 'lib'
  end
  command "/usr/local/bin/apache2_module_conf_generate.pl /usr/#{libdir}/httpd/
modules /etc/httpd/mods-available"
  action :run
end
```

La ressource détermine d'abord le type d'UC. `[:kernel][:machine]` est un autre des attributs automatiques que Chef génère pour représenter les différentes propriétés système, le type d'UC dans le cas présent. Ensuite, elle spécifie la commande, un script Perl et utilise une action `run` pour exécuter le script, qui génère les fichiers module.load. Pour plus d'informations, consultez execute.

Une `template` ressource représente un fichier, généralement un fichier de configuration, qui doit être généré à partir de l'un des fichiers modèles du livre de recettes. L'exemple suivant crée un fichier de configuration `httpd.conf` à partir du modèle `apache2.conf.erb` qui a été présenté dans Modèles.

```
template 'apache2.conf' do
  case node[:platform]
  when 'centos','redhat','fedora','amazon'
    path "#{node[:apache][:dir]}/conf/httpd.conf"
  when 'debian','ubuntu'
    path "#{node[:apache][:dir]}/apache2.conf"
  end
  source 'apache2.conf.erb'
  owner 'root'
  group 'root'
  mode 0644
  notifies :restart, resources(:service => 'apache2')
end
```

La ressource détermine le nom et l'emplacement du fichier généré en fonction du système d'exploitation de l'instance. Elle spécifie ensuite `apache2.conf.erb` en tant que modèle à utiliser pour générer le fichier et définit le propriétaire du fichier, le groupe et le mode. Elle exécute l'action `notify` pour demander à la ressource `service` qui représente le serveur Apache de redémarrer le serveur. Pour plus d'informations, consultez template.

Attributs de déploiement et de configuration de pile : Linux

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette rubrique inclut les attributs de configuration et de déploiement de pile les plus couramment utilisés et leur syntaxe nœud associée. Elle est organisée autour de la structure d'espace de noms de configuration de pile utilisée par les piles Linux. Notez que les mêmes noms d'attribut sont parfois utilisés à différentes fins, et qu'ils interviennent dans des espaces de noms différents. Par exemple, `id` peut faire référence à un ID de pile, un ID de couche, un ID d'application, etc., et que, par conséquent, vous avez besoin du nom complet pour utiliser la valeur de l'attribut. Une solution pratique consiste à visualiser ces données comme objet JSON. Pour obtenir des exemples, consultez Attributs de déploiement et de configuration de pile.

> **Note**
>
> Sur les instances Linux, AWS OpsWorks Stacks installe cet objet JSON sur chaque instance en plus d'ajouter les données à l'objet nœud. Vous pouvez le récupérer avec la commande `get_json` de l'interface de ligne de commande de l'agent.

Rubriques

- Attributs opsworks
- Attributs opsworks_custom_cookbooks
- Attributs dependencies
- Attributs ganglia
- Attributs mysql
- Attributs passenger
- Attributs opsworks_bundler
- Attributs deploy

- Autres attributs de niveau supérieur

Attributs opsworks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

L'`opsworks`élément, parfois appelé espace de `opsworks` noms, contient un ensemble d'attributs qui définissent la configuration de base de la pile.

> **Important**
>
> Il n'est pas recommandé de remplacer les valeurs d'attribut dans l'espace de noms `opsworks`. Cette action peut entraîner un échec des recettes intégrées.

Rubriques

- applications
- Attributs instance
- Attributs layers
- Attributs rails_stack
- Attributs stack
- Autres attributs opsworks de niveau supérieur

applications

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement

aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Contient une liste d'objets incorporés, un pour chaque application de la pile. Chaque objet incorporé contient les attributs suivants qui décrivent la configuration de l'application.

> **Note**
>
> La syntaxe nœud générale de ces attributs est la suivante, où *i* spécifie l'index de liste de base zéro de l'instance.
>
> ```
> node["opsworks"]["applications"]["i"]["attribute_name"]
> ```

application_type

Type de l'application (chaîne). Les valeurs possibles sont les suivantes :

- `php` : application PHP
- `rails` : application Ruby on Rails
- `java` : application Java
- `nodejs` : application Node.js
- `web` : page HTML statique
- `other` : tous les autres types d'applications

```
node["opsworks"]["applications"]["i"]["application_type"]
```

name

Nom complet défini par l'utilisateur, tel que `"SimplePHP"` (chaîne).

```
node["opsworks"]["applications"]["i"]["name"]
```

slug_name

Un nom court, qui est un nom entièrement en minuscules, tel `"simplephp"` que celui généré par le nom OpsWorks de l'application (chaîne).

```
node["opsworks"]["applications"]["i"]["slug_name"]
```

Attributs instance

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

L'attribut `instance` contient un ensemble d'attributs qui spécifient la configuration de l'instance.

application	availability_zone	backends
aws_instance_id	hostname	id
instance_type	ip	couches
private_dns_name	private_ip	public_dns_name
region		

application

Architecture de l'instance, telle que `"i386"` (chaîne).

```
node["opsworks"]["instance"]["architecture"]
```

availability_zone

Zone de disponibilité de l'instance, telle que `"us-west-2a"` (chaîne).

```
node["opsworks"]["instance"]["availability_zone"]
```

backends

Nombre de processus web principaux (chaîne). Il détermine, par exemple, le nombre de connexions simultanées qui HAProxy seront transmises à un backend Rails. La valeur par défaut dépend de la mémoire de l'instance et du nombre de cœurs.

```
node["opsworks"]["instance"]["backends"]
```

aws_instance_id

L'ID de l' EC2 instance (chaîne).

```
node["opsworks"]["instance"]["aws_instance_id"]
```

hostname

Nom d'hôte, tel que "`php-app1`" (chaîne).

```
node["opsworks"]["instance"]["hostname"]
```

id

L'ID d'instance, qui est un GUID AWS OpsWorks généré par Stacks qui identifie l'instance de manière unique (chaîne).

```
node["opsworks"]["instance"]["id"]
```

instance_type

Type de l'instance, tel que "`c1.medium`" (chaîne).

```
node["opsworks"]["instance"]["instance_type"]
```

ip

Adresse IP publique (chaîne).

```
node["opsworks"]["instance"]["ip"]
```

couches

Liste des couches de l'instance, identifiées par leurs noms courts, comme `"lb"` ou `"db-master"` (liste de chaînes).

```
node["opsworks"]["instance"]["layers"]
```

private_dns_name

Nom DNS privé (chaîne).

```
node["opsworks"]["instance"]["private_dns_name"]
```

private_ip

Adresse IP privée (chaîne).

```
node["opsworks"]["instance"]["private_ip"]
```

public_dns_name

Nom DNS public (chaîne).

```
node["opsworks"]["instance"]["public_dns_name"]
```

region

Région AWS, telle que `"us-west-2"` (chaîne).

```
node["opsworks"]["instance"]["region"]
```

Attributs layers

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

L'attribut `layers` contient un ensemble d'attributs de couche, un pour chacune des couches de la pile et nommés avec le nom court de la couche, tel que `php-app`. Une pile peut avoir au plus l'une des couches intégrées, dont les noms courts sont les suivants :

- `db-master`: couche MySQL
- `java-app`: couche Java App Server
- `lb`: HAProxy couche
- `monitoring-master`: couche ganglionnaire
- `memcached`: couche Memcached
- `nodejs-app`: couche de serveur d'applications Node.js
- `php-app`: couche de serveur d'applications PHP
- `rails-app`: couche Rails App Server
- `web`: couche de serveur Web statique

Une pile peut contenir un nombre quelconque de couches personnalisées, lesquelles ont des noms courts définis par l'utilisateur.

Chaque attribut de couche contient les attributs suivants :

- id
- instances
- nom

id

L'ID de couche, qui est un GUID généré par OpsWorks et identifie de manière unique la couche (chaîne).

```
node["opsworks"]["layers"]["layershortname"]["id"]
```

Instances

L'élément `instances` contient un ensemble d'attributs d'instance, un pour chacune des instances en ligne de la couche. Ils sont nommés avec le nom d'hôte de l'instance, tel que `php-app1`.

> **Note**
>
> L'élément `instances` contient uniquement les instances qui sont dans l'état en ligne lorsque les attributs de configuration et de déploiement de pile sont créés.

Chaque élément d'instance contient les attributs suivants :

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

Zone de disponibilité, telle que `"us-west-2a"` (chaîne).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["availability_zone"]
```

aws_instance_id

L'ID de l' EC2 instance (chaîne).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["aws_instance_id"]
```

backends

Nombre de processus web principaux (nombre). Il détermine, par exemple, le nombre de connexions simultanées qui HAProxy seront transmises à un backend Rails. La valeur par défaut dépend de la mémoire de l'instance et du nombre de cœurs.

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["backends"]
```

booted_at

Heure à laquelle l' EC2 instance a été démarrée, en utilisant le format UTC:MM:SS+HH:mm yyyy-mm-dddThh (chaîne). Par exemple, `"2013-10-01T08:35:22+00:00"` correspond à 08:35:22 le 10 octobre 2013, sans décalage horaire. Pour plus d'informations, consultez ISO 8601.

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["booted_at"]
```

created_at

Heure à laquelle l' EC2 instance a été créée, en utilisant le format UTC:MM:SS+HH:mm yyyy-mm-dddThh (chaîne). Par exemple, `"2013-10-01T08:35:22+00:00"` correspond à 08:35:22 le 10 octobre 2013, sans décalage horaire. Pour plus d'informations, consultez ISO 8601.

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["created_at"]
```

elastic_ip

Adresse IP Elastic, qui possède la valeur null si l'instance n'a pas d'adresse (chaîne).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["elastic_ip"]
```

instance_type

Type de l'instance, tel que `"c1.medium"` (chaîne).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["instance_type"]
```

ip

Adresse IP publique (chaîne).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["ip"]
```

private_ip

Adresse IP privée (chaîne).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["private_ip"]
```

public_dns_name

Nom DNS public (chaîne).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["public_dns_name"]
```

private_dns_name

Nom DNS privé (chaîne).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["private_dns_name"]
```

region

Région AWS, telle que `"us-west-2"` (chaîne).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["region"]
```

status

État (chaîne). Les valeurs possibles sont les suivantes :

- "requested"
- "booting"
- "running_setup"
- "online"
- "setup_failed"
- "start_failed"
- "terminating"
- "terminated"
- "stopped"
- "connection_lost"

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["status"]
```

name

Nom de la couche, utilisé pour représenter la couche dans la console (chaîne). Il peut être défini par l'utilisateur et n'est pas nécessairement unique.

```
node["opsworks"]["layers"]["layershortname"]["name"]
```

Attributs rails_stack

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

name

Spécifie la pile de rails et est défini sur "apache_passenger" ou "nginx_unicorn" (chaîne).

```
node["opsworks"]["rails_stack"]["name"]
```

recipe

Recette associée, qui varie selon que vous utilisez Passenger ou Unicorn (chaîne) :

- Unicorn : `"unicorn::rails"`
- Passenger : `"passenger_apache2::rails"`

```
node["opsworks"]["rails_stack"]["recipe"]
```

restart_command

Commande de redémarrage, qui varie selon que vous utilisez Passenger ou Unicorn (chaîne) :

- Unicorn : `"../../shared/scripts/unicorn clean-restart"`
- Passenger : `"touch tmp/restart.txt"`

web

Nom de service, qui varie selon que vous utilisez Passenger ou Unicorn (chaîne) :

- Unicorn : `"unicorn"`
- Passenger : `"apache2"`

```
node["opsworks"]["rails_stack"]["service"]
```

Attributs stack

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les attributs `stack` spécifient certains aspects de la configuration de la pile, telles que les configurations de la couche de service.

- elb-load-balancers
- id
- name
- rds_instances
- vpc_id

elb-load-balancers

Contient la liste des objets incorporés, un par équilibreur de charge Elastic Load Balancing de la pile. Chaque objet incorporé contient les attributs suivants qui décrivent la configuration de l'équilibreur de charge.

> **Note**
>
> La syntaxe nœud générale de ces attributs est la suivante, où *i* spécifie l'index de liste de base zéro de l'instance.
>
> ```
> node["opsworks"]["stack"]["elb-load-balancers"]["i"]["attribute_name"]
> ```

dns_name

Nom DNS de l'équilibreur de charge (chaîne).

```
node["opsworks"]["stack"]["elb-load-balancers"]["i"]["dns_name"]
```

name

Nom de l'équilibreur de charge (chaîne).

```
node["opsworks"]["stack"]["elb-load-balancers"]["i"]["name"]
```

layer_id

ID de la couche à laquelle l'équilibreur de charge est attaché (chaîne).

```
node["opsworks"]["stack"]["elb-load-balancers"]["i"]["layer_id"]
```

id

ID de la pile (chaîne).

```
node["opsworks"]["stack"]["id"]
```

name

Nom de la pile (chaîne).

```
node["opsworks"]["stack"]["name"]
```

rds_instances

Contient la liste des objets incorporés, un par instance Amazon RDS enregistrée auprès de la pile. Chaque objet incorporé contient un ensemble d'attributs qui définissent la configuration de l'instance. Vous spécifiez ces valeurs lorsque vous utilisez l'API ou la console Amazon RDS pour créer l'instance. Vous pouvez également utiliser la console ou l'API Amazon RDS pour modifier certains paramètres une fois l'instance créée. Pour plus d'informations, consultez la documentation Amazon RDS.

> **Note**
>
> La syntaxe nœud générale de ces attributs est la suivante, où *i* spécifie l'index de liste de base zéro de l'instance.
>
> ```
> node["opsworks"]["stack"]["rds_instances"]["i"]["attribute_name"]
> ```

Si votre stack comporte plusieurs instances Amazon RDS, voici un exemple d'utilisation d'une instance particulière dans une recette.

```
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
```

adresse	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	

adresse

URL d'instances, telle que `opsinstance.ccdvt3hwog1a.us-west-2.rds.amazonaws.com` (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["address"]
```

allocated_storage

Espace de stockage alloué, en Go (nombre).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["allocated_storage"]
```

arn

ARN de l'instance (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["arn"]
```

auto_minor_version_upgrade

Indique si les mises à niveau de version mineure doivent être appliquées automatiquement (valeur booléenne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["auto_minor_version_upgrade"]
```

availability_zone

Zone de disponibilité de l'instance, telle que `us-west-2a` (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["availability_zone"]
```

backup_retention_period

Période de conservation des sauvegardes, en jours (nombre).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["backup_retention_period"]
```

db_instance_class

Classe d'instances de base de données, telle que `db.m1.small` (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_instance_class"]
```

db_instance_identifier

Identifiant d'instance DB défini par l'utilisateur (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_instance_identifier"]
```

db_instance_status

État de l'instance (chaîne). Pour plus d'informations, consultez Instance de base de données.

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_instance_status"]
```

db_name

Nom de base de données défini par l'utilisateur (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_name"]
```

db_parameter_groups

Groupes de paramètres DB de l'instance, qui contient une liste d'objets incorporés, un pour chaque groupe de paramètres. Pour plus d'informations, consultez Utilisation des groupes de paramètres de base de données. Chaque objet contient les attributs suivants :

db_parameter_group_name

Nom du groupe (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_parameter_groups"][j"]
["db_parameter_group_name"]
```

parameter_apply_status

État de l'application (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_parameter_groups"][j"]
["parameter_apply_status"]
```

db_security_groups

Groupes de sécurité DB de l'instance, qui contient une liste d'objets incorporés, un par groupe de sécurité. Pour plus d'informations, consultez Utilisation des groupes de sécurité DB. Chaque objet contient les attributs suivants :

db_security_group_name

Nom du groupe de sécurité (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_security_groups"][j"]
["db_security_group_name"]
```

status

État (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_security_groups"][j"]
["status"]
```

db_user

Nom d'utilisateur principal défini par l'utilisateur (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_user"]
```

engine

Moteur de la base de données, tel que `mysql(5.6.13)` (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["engine"]
```

instance_create_time

Date de création de l'instance, telle que `2014-04-15T16:13:34Z` (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["instance_create_time"]
```

license_model

Modèle de licence de l'instance, tel que `general-public-license` (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["license_model"]
```

multi_az

Indique si le déploiement multi-AZ est activé (valeur booléenne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["multi_az"]
```

option_group_memberships

Appartenances des groupes d'options de l'instance, qui contient une liste d'objets incorporés, un par groupe d'options. Pour plus d'informations, consultez Utilisation des groupes d'options. Chaque objet contient les attributs suivants :

option_group_name

Nom du groupe (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["option_group_memberships"]
[j"]["option_group_name"]
```

status

État du groupe (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["option_group_memberships"]
[j"]["status"]
```

port

Port du serveur de base de données (nombre).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["port"]
```

preferred_backup_window

Fenêtre de sauvegarde quotidienne préférée, telle que `06:26-06:56` (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["preferred_backup_window"]
```

preferred_maintenance_window

Fenêtre de maintenance hebdomadaire préférée, telle que `thu:07:13-thu:07:43` (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["preferred_maintenance_window"]
```

publicly_accessible

Indique si la base de données est accessible publiquement (valeur booléenne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["publicly_accessible"]
```

read_replica_db_instance_identifiers

Liste d'identifiants d'instance de réplica en lecture (liste de chaînes). Pour plus d'informations, consultez Utilisation des réplicas en lecture.

```
node["opsworks"]["stack"]["rds_instances"]["i"]
["read_replica_db_instance_identifiers"]
```

region

Région AWS, telle que `us-west-2` (chaîne).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["region"]
```

status_infos

Liste d'informations d'état (liste de chaînes).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["status_infos"]
```

vpc_security_groups

Liste de groupes de sécurité VPC (liste de chaînes).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["vpc_security_groups"]
```

vpc_id

ID VPC (chaîne). La valeur est `null` si l'instance n'est pas dans un VPC.

```
node["opsworks"]["stack"]["vpc_id"]
```

Autres attributs opsworks de niveau supérieur

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section contient les attributs `opsworks` qui n'ont pas d'attributs enfants.

activity

Activité qui est associée aux attributs, telle que `deploy` (chaîne).

```
node["opsworks"]["activity"]
```

agent_version

Version de l' OpsWorks agent de l'instance (chaîne).

```
node["opsworks"]["agent_version"]
```

deploy_chef_provider

Fournisseur de déploiement Chef, qui influe sur la structure de répertoire d'une application déployée (chaîne). Vous pouvez attribuer à cet attribut l'une des valeurs suivantes :

- `Branch`
- `Revision`
- `Timestamped` (valeur par défaut)

```
node["opsworks"]["deploy_chef_provider"]
```

ruby_stack

Pile Ruby (chaîne). Le paramètre par défaut est la version « enterprise » (`ruby_enterprise`). Pour la version MRI, définissez l'attribut avec la valeur `ruby`.

```
node["opsworks"]["ruby_stack"]
```

ruby_version

Version Ruby qui sera utilisée par les applications (chaîne). Vous pouvez utiliser cet attribut pour spécifier uniquement les versions majeure et mineure. Vous devez utiliser l'attribut [["ruby"]] approprié pour spécifier la version du correctif. Pour plus d'informations sur la façon de spécifier une version, exemples inclus, consultez Versions de Ruby. Pour obtenir tous les détails sur la façon dont AWS OpsWorks Stacks détermine la version Ruby, consultez le fichier d'attributs intégrés, ruby.rb.

```
node["opsworks"]["ruby_version"]
```

run_cookbook_tests

S'il faut exécuter minitest-chef-handlerdes tests sur vos livres de recettes Chef 11.4 (booléen).

```
node["opsworks"]["run_cookbook_tests"]
```

sent_at

Indique quand la commande a été envoyée à l'instance (nombre).

```
node["opsworks"]["sent_at"]
```

déploiement

Si ces attributs sont associés à une activité de déploiement, `deployment` est défini comme l'ID de déploiement, un GUID généré par AWS OpsWorks Stacks qui identifie de façon unique le déploiement (chaîne). Sinon, l'attribut est défini sur null.

```
node["opsworks"]["deployment"]
```

Attributs opsworks_custom_cookbooks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Contient les attributs qui spécifient les livres personnalisés de la pile.

activé

Indique si les livres personnalisés sont activés (valeur booléenne).

```
node["opsworks_custom_cookbooks"]["enabled"]
```

recipes

Liste des recettes qui doivent être exécutées pour cette commande, y compris les recettes personnalisées, à l'aide du format *cookbookname*::*recipename* (liste de chaînes).

```
node["opsworks_custom_cookbooks"]["recipes"]
```

Attributs dependencies

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Contient plusieurs attributs qui sont liés à la `update_dependencies`commande de pileExécution des commandes de pile.

gem_binary

Emplacement des fichiers binaires Gem (chaîne).

upgrade_debs

Indique s'il faut mettre à niveau les packages Debs (valeur booléenne).

update_debs

Indique s'il faut mettre à jour les packages Debs (valeur booléenne).

Attributs ganglia

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Contient un attribut web comportant plusieurs attributs qui spécifient comment accéder à la page web des statistiques Ganglia :

mot de passe

Mot de passe requis pour accéder à la page des statistiques (chaîne).

```
node["ganglia"]["web"]["password"]
```

url

Chemin de l'URL de la page des statistiques, tel que `"/ganglia"` (chaîne). L'URL complète est `http://`*`DNSNameURLPath`*, où *`DNSName`* est le nom DNS de l'instance associée.

```
node["ganglia"]["web"]["url"]
```

utilisateur

Nom d'utilisateur requis pour accéder à la page des statistiques (chaîne).

```
node["ganglia"]["web"]["user"]
```

Attributs mysql

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Contient un ensemble d'attributs qui spécifient la configuration du serveur de base de données MySQL.

clients

Liste d'adresses IP de clients (liste de chaînes).

```
node["mysql"]["clients"]
```

server_root_password

Mot de passe racine (chaîne).

```
node["mysql"]["server_root_password"]
```

Attributs passenger

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Contient un ensemble d'attributs qui spécifient la configuration Phusion Passenger.

gem_bin

L'emplacement des RubyGems fichiers binaires, par exemple "`/usr/local/bin/gem`" (chaîne).

```
node["passenger"]["gem_bin"]
```

max_pool_size

Taille de pool maximale (nombre).

```
node["passenger"]["max_pool_size"]
```

ruby_bin

Emplacement des fichiers binaires Ruby, comme "`/usr/local/bin/ruby`".

```
node["passenger"]["ruby_bin"]
```

version

Version Passenger (chaîne).

```
node["passenger"]["version"]
```

Attributs opsworks_bundler

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Contient les éléments qui spécifient la prise en charge Bundler.

manage_package

Indique s'il convient d'installer et de gérer Bundler (valeur booléenne).

```
node["opsworks_bundler"]["manage_package"]
```

version

Version Bundler (chaîne).

```
node["opsworks_bundler"]["version"]
```

Attributs deploy

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Si les attributs sont associés à un événement Deploy ou à une commande de pile Execute Recipes, l'attribut `deploy` contient un attribut pour chaque application déployée, nommé d'après le nom court de l'application. Chaque attribut d'application contient les attributs suivants :

application	application_type	auto_bundle_on_deploy
database	deploy_to	domains
document_root	environment_variables	groupe
keep_releases	memcached	migrate
mounted_at	purge_before_symlink	rails_env
restart_command	scm	ssl_certificate
ssl_certificate_ca	ssl_certificate_key	ssl_support
pile	symlink_before_migrate	symlinks
utilisateur		

application

Nom slug de l'application, tel que `"simplephp"` (chaîne).

```
node["deploy"]["appshortname"]["application"]
```

application_type

Type d'application (chaîne). Les valeurs possibles sont les suivantes :

- `java` : application Java
- `nodejs` : application Node.js
- `php` : application PHP
- `rails` : application Ruby on Rails
- `web` : page HTML statique
- `other` : tous les autres types d'applications

```
node["deploy"]["appshortname"]["application_type"]
```

auto_bundle_on_deploy

Pour les applications Rails, indique s'il faut exécuter Bundler au cours du déploiement (valeur booléenne).

```
node["deploy"]["appshortname"]["auto_bundle_on_deploy"]
```

database

Contient les informations requises pour la connexion à la base de données de l'application. Si une couche de base de données est attachée à l'application, AWS OpsWorks Stacks attribue automatiquement les valeurs appropriées à ces attributs.

adapter

Adaptateur de base de données, tel que `mysql` (chaîne).

```
node["deploy"]["appshortname"]["database"]["adapter"]
```

database

Nom de la base de données, qui est généralement le nom slug de l'application, tel que `"simplephp"` (chaîne).

```
node["deploy"]["appshortname"]["database"]["database"]
```

data_source_provider

Source de données : `mysql` ou `rds` (chaîne).

```
node["deploy"]["appshortname"]["database"]["data_source_provider"]
```

hôte

Adresse IP de l'hôte de base de données (chaîne).

```
node["deploy"]["appshortname"]["database"]["host"]
```

mot de passe

Mot de passe de la base de données (chaîne).

```
node["deploy"]["appshortname"]["database"]["password"]
```

port

Port de la base de données (nombre).

```
node["deploy"]["appshortname"]["database"]["port"]
```

reconnect

Pour les applications Rails, indique si l'application doit se reconnecter si la connexion n'existe plus (valeur booléenne).

```
node["deploy"]["appshortname"]["database"]["reconnect"]
```

nom d'utilisateur

Nom d'utilisateur (chaîne).

```
node["deploy"]["appshortname"]["database"]["username"]
```

deploy_to

Indique où l'application doit être déployée, par exemple, `"/srv/www/simplephp"` (chaîne).

```
node["deploy"]["appshortname"]["deploy_to"]
```

domains

Liste des domaines de l'application (liste de chaînes).

```
node["deploy"]["appshortname"]["domains"]
```

document_root

Racine du document, si vous spécifiez une racine personnalisée ou null si vous utilisez la racine par défaut (chaîne).

```
node["deploy"]["appshortname"]["document_root"]
```

environment_variables

Collection de vingt attributs au plus qui représentent les variables d'environnement spécifiées par l'utilisateur et qui ont été définies pour l'application. Pour plus d'informations sur la définition des variables d'environnement d'une application, consultez Ajout d'applications. Comme chaque nom d'attribut est défini sur un nom de variable d'environnement et que la valeur correspondante est définie sur la valeur de la variable, vous pouvez utiliser la syntaxe suivante pour faire référence à une valeur particulière.

```
node["deploy"]["appshortname"]["environment_variables"]["variable_name"]
```

groupe

Groupe de l'application (chaîne).

```
node["deploy"]["appshortname"]["group"]
```

keep_releases

Nombre de déploiements d'applications que AWS OpsWorks Stacks stockera (nombre). Cet attribut contrôle le nombre de fois que vous pouvez restaurer une application. Par défaut, l'attribut est défini sur la valeur globale, deploy_keep_releases , qui possède une valeur par défaut de 5. Vous pouvez remplacer `keep_releases` pour spécifier le nombre de déploiements stockés d'une application spécifique.

```
node["deploy"]["appshortname"]["keep_releases"]
```

memcached

Contient deux attributs qui définissent la configuration memcached.

hôte

Adresse IP (chaîne) de l'instance du serveur Memcached.

```
node["deploy"]["appshortname"]["memcached"]["host"]
```

port

Port sur lequel écoute le serveur memcached (nombre).

```
node["deploy"]["appshortname"]["memcached"]["port"]
```

migrate

Pour les applications Rails, indique s'il convient d'exécuter les migrations (valeur booléenne).

```
node["deploy"]["appshortname"]["migrate"]
```

mounted_at

Point de montage de l'application, si vous spécifiez un point de montage personnalisé ou null si vous utilisez le point de montage par défaut (chaîne).

```
node["deploy"]["appshortname"]["mounted_at"]
```

purge_before_symlink

Pour les applications Rails, le tableau des chemins d'accès doit être effacé avant de créer les liens symboliques (liste de chaînes).

```
node["deploy"]["appshortname"]["purge_before_symlink"]
```

rails_env

Pour les instances de Rails App Server, l'environnement des rails, tel que `"production"` (chaîne).

```
node["deploy"]["appshortname"]["rails_env"]
```

restart_command

Commande à exécuter au redémarrage de l'application, telle que `"echo 'restarting app'"`.

```
node["deploy"]["appshortname"]["restart_command"]
```

scm

Contient un ensemble d'attributs qui spécifient les informations OpsWorks utilisées pour déployer l'application à partir de son référentiel de contrôle de source. Les attributs varient en fonction du type de référentiel.

mot de passe

Mot de passe pour les référentiels privés et null pour les référentiels publics (chaîne). Pour les compartiments Amazon S3 privés, l'attribut est défini sur la clé secrète.

```
node["deploy"]["appshortname"]["scm"]["password"]
```

référentiels

URL de référentiel, telle que `"git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git"` (chaîne).

```
node["deploy"]["appshortname"]["scm"]["repository"]
```

révision

Si le référentiel comporte plusieurs succursales, l'attribut spécifie la branche ou la version de l'application, telle que `"version1"` (chaîne). Sinon, l'attribut a la valeur null.

```
node["deploy"]["appshortname"]["scm"]["revision"]
```

scm_type

Type de référentiel (chaîne). Les valeurs possibles sont les suivantes :

- `"git"` : référentiel Git
- `"svn"` : référentiel Subversion

- `"s3"`: un compartiment Amazon S3
- `"archive"` : archive HTTP
- `"other"` : autre type de référentiel

```
node["deploy"]["appshortname"]["scm"]["scm_type"]
```

ssh_key

Clé SSH de déploiement, pour accéder aux référentiels Git privés et null pour les référentiels publics (chaîne).

```
node["deploy"]["appshortname"]["scm"]["ssh_key"]
```

utilisateur

Nom d'utilisateur pour les référentiels privés et null pour les référentiels publics (chaîne). Pour les compartiments Amazon S3 privés, l'attribut est défini sur la clé d'accès.

```
node["deploy"]["appshortname"]["scm"]["user"]
```

ssl_certificate

Certificat SSL de l'application, si vous avez activé la prise en charge SSL, ou null sinon (chaîne).

```
node["deploy"]["appshortname"]["ssl_certificate"]
```

ssl_certificate_ca

Si SSL est activé, attribut pour spécifier une clé d'autorité de certification intermédiaire ou une authentification client (chaîne).

```
node["deploy"]["appshortname"]["ssl_certificate_ca"]
```

ssl_certificate_key

Clé privée SSL de l'application, si vous avez activé la prise en charge SSL, ou null sinon (chaîne).

```
node["deploy"]["appshortname"]["ssl_certificate_key"]
```

ssl_support

Indique si SSL est pris en charge (valeur booléenne).

```
node["deploy"]["appshortname"]["ssl_support"]
```

pile

Contient un attribut booléen, `needs_reload`, qui spécifie s'il faut recharger le serveur d'applications durant le déploiement.

```
node["deploy"]["appshortname"]["stack"]["needs_reload"]
```

symlink_before_migrate

Pour les applications Rails, contient les liens symboliques qui doivent être créés avant d'exécuter les migrations en tant que paires `"link":"target"`.

```
node["deploy"]["appshortname"]["symlink_before_migrate"]
```

symlinks

Contient les liens symboliques du déploiement sous forme de paires `"link":"target"`.

```
node["deploy"]["appshortname"]["symlinks"]
```

utilisateur

Utilisateur de l'application (chaîne).

```
node["deploy"]["appshortname"]["user"]
```

Autres attributs de niveau supérieur

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section contient les attributs de configuration de pile de niveau supérieur qui n'ont pas d'attributs enfants.

Attributs rails

Contient un attribut max_pool_size qui spécifie la taille de pool maximale du serveur (nombre). La valeur de l'attribut est définie par AWS OpsWorks Stacks et dépend du type d'instance, mais vous pouvez la remplacer en utilisant du JSON personnalisé ou un fichier d'attribut personnalisé.

```
node["rails"]["max_pool_size"]
```

Attributs recipes

Liste de recettes intégrées exécutées par cette activité, selon le format "*cookbookname*::*recipename*" (liste de chaînes).

```
node["recipes"]
```

Attributs opsworks_rubygems

Contient un élément de version qui spécifie la RubyGems version (chaîne).

```
node["opsworks_rubygems"]["version"]
```

Attributs languages

Contient un attribut pour chaque langue installée, nommé d'après la langue, tel que ruby. L'attribut est un objet qui contient un attribut, tel que ruby_bin, qui spécifie le dossier d'installation, tel que "/usr/bin/ruby" (chaîne).

Attributs ssh_users

Contient un ensemble d'attributs, dont chacun décrit l'un des utilisateurs auxquels des autorisations SSH ont été accordées. Chaque attribut est nommé avec l'identifiant Unix d'un utilisateur. AWS OpsWorks Stacks génère un identifiant unique pour chaque utilisateur compris entre 2000 et 4000, par exemple"2001", et crée un utilisateur avec cet identifiant sur chaque

instance. Dans la AWS OpsWorks mesure où les réserves se situent entre 2000 et 4000, les utilisateurs que vous créez en dehors AWS OpsWorks (en utilisant des recettes de livres de recettes ou en important des utilisateurs AWS OpsWorks depuis IAM, par exemple) peuvent se UIDs faire remplacer par AWS OpsWorks Stacks pour un autre utilisateur. Il est recommandé de créer des utilisateurs et de gérer leur accès dans la console AWS OpsWorks Stacks. Si vous créez des utilisateurs en dehors de AWS OpsWorks Stacks, utilisez des *UnixID* valeurs supérieures à 4 000.

Chaque attribut contient les attributs suivants :

e-mail

Adresse de messagerie de l'utilisateur (chaîne).

```
node["ssh_users"]["UnixID"]["email"]
```

public_key

Clé SSH publique de l'utilisateur (chaîne).

```
node["ssh_users"]["UnixID"]["public_key"]
```

sudoer

Indique si l'utilisateur a les autorisation sudo (valeur booléenne).

```
node["ssh_users"]["UnixID"]["sudoer"]
```

name

Nom d'utilisateur (chaîne).

```
node["ssh_users"]["UnixID"]["name"]
```

Attributs des livres de recettes intégrés

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement

aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> La plupart de ces attributs ne sont disponibles que sur les piles Linux.

La plupart des recettes intégrées ont un ou plusieurs fichiers d'attributs qui définissent différents paramètres. Vous pouvez accéder à ces paramètres dans vos recettes personnalisées et utiliser un JSON personnalisé pour les remplacer. Vous devez généralement accéder aux attributs qui contrôlent la configuration des différentes technologies de serveur prises en charge par AWS OpsWorks Stacks ou les remplacer. Cette section résume ces attributs. Les fichiers d'attributs complets, ainsi que les recettes et modèles associés, sont disponibles sur https://github.com/aws/opsworks-cookbooks.git.

> **Note**
>
> Tous les attributs de recette intégrés sont de type `default`.

Rubriques

- Attributs apache2
- Attributs deploy
- Attributs haproxy
- Attributs memcached
- Attributs mysql
- Attributs nginx
- Attributs opsworks_berkshelf
- Attributs opsworks_java
- Attributs passenger_apache2
- Attributs ruby

- Attributs unicorn

Attributs apache2

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Ces attributs ne sont disponibles que sur les piles Linux.

Les attributs apache2 spécifient la configuration du serveur HTTP Apache. Pour plus d'informations, consultez Fonctionnalités principales Apache. Pour plus d'informations sur la façon de remplacer les attributs intégrés afin de spécifier les valeurs personnalisées, consultez Remplacement des attributs.

binary	contact	deflate_types
dir	document_root	groupe
hide_info_headers	icondir	init_script
keepalive	keepaliverequests	keepalivetimeout
lib_dir	libexecdir	listen_ports
log_dir	Attributs logrotate	pid_file
Attributs prefork	serversignature	servertokens
timeout	traceenable	utilisateur
version	Attributs worker	

binary

Emplacement du fichier binaire Apache (chaîne). La valeur par défaut est `'/usr/sbin/httpd'`.

```
node[:apache][:binary]
```

contact

Contact e-mail (chaîne). La valeur par défaut est une adresse fictive, `'ops@example.com'`.

```
node[:apache][:contact]
```

deflate_types

Demande à `mod_deflate` d'activer la compression pour les types Mime spécifiés, s'ils sont pris en charge par le navigateur (liste de chaînes). La valeur par défaut est la suivante :

```
['application/javascript',
 'application/json',
 'application/x-javascript',
 'application/xhtml+xml',
 'application/xml',
 'application/xml+rss',
 'text/css',
 'text/html',
 'text/javascript',
 'text/plain',
 'text/xml']
```

> **Warning**
>
> La compression peut entraîner des risques de sécurité. Pour désactiver complètement la compression, définissez l'attribut comme suit :
>
> ```
> node[:apache][:deflate_types] = []
> ```

```
node[:apache][:deflate_types]
```

dir

Répertoire racine du serveur (chaîne). Les valeurs par défaut sont les suivantes :

- Amazon Linux et Red Hat Enterprise Linux (RHEL) : `'/etc/httpd'`
- Ubuntu : `'/etc/apache2'`

```
node[:apache][:dir]
```

document_root

Racine du document (chaîne). Les valeurs par défaut sont les suivantes :

- Amazon Linux et RHEL : `'/var/www/html'`
- Ubuntu : `'/var/www'`

```
node[:apache][:document_root]
```

groupe

Nom du groupe (chaîne). Les valeurs par défaut sont les suivantes :

- Amazon Linux et RHEL : `'apache'`
- Ubuntu : `'www-data'`

```
node[:apache][:group]
```

hide_info_headers

Indique s'il convient d'ignorer les informations de version et de module des en-têtes HTTP (`'true'`/`'false'`) (chaîne). La valeur par défaut est `'true'`.

```
node[:apache][:hide_info_headers]
```

icondir

Répertoire des icônes (chaîne). Les valeurs par défaut sont les suivantes :

- Amazon Linux et RHEL : `'/var/www/icons/'`
- Ubuntu : `'/usr/share/apache2/icons'`

```
node[:apache][:icondir]
```

init_script

Script d'initialisation (chaîne). Les valeurs par défaut sont les suivantes :

- Amazon Linux et RHEL : `'/etc/init.d/httpd'`
- Ubuntu : `'/etc/init.d/apache2'`

```
node[:apache][:init_script]
```

keepalive

Indique si vous voulez que les connexions demeurent permanentes (chaîne). Les valeurs possibles sont `'On'` et `'Off'` (chaîne). La valeur par défaut est `'Off'`.

```
node[:apache][:keepalive]
```

keepaliverequests

Nombre maximal de requêtes toujours actives qu'Apache gère en même temps (nombre). La valeur par défaut est `100`.

```
node[:apache][:keepaliverequests]
```

keepalivetimeout

Durée pendant laquelle Apache attend une demande avant de fermer la connexion (nombre). La valeur par défaut est `3`.

```
node[:apache][:keepalivetimeout]
```

lib_dir

Répertoire qui contient les bibliothèques de code objet (chaîne). Les valeurs par défaut sont les suivantes :

- Amazon Linux (x86) : `'/usr/lib/httpd'`
- Amazon Linux (x64) et RHEL : `'/usr/lib64/httpd'`
- Ubuntu : `'/usr/lib/apache2'`

```
node[:apache][:lib_dir]
```

libexecdir

Répertoire qui contient les fichiers exécutables du programme (chaîne). Les valeurs par défaut sont les suivantes :

- Amazon Linux (x86) : `'/usr/lib/httpd/modules'`
- Amazon Linux (x64) et RHEL : `'/usr/lib64/httpd/modules'`
- Ubuntu : `'/usr/lib/apache2/modules'`

```
node[:apache][:libexecdir]
```

listen_ports

Liste des ports sur lesquels le serveur écoute (liste de chaînes). La valeur par défaut est `[ '80','443' ]`.

```
node[:apache][:listen_ports]
```

log_dir

Répertoire des journaux (chaîne). Les valeurs par défaut sont les suivantes :

- Amazon Linux et RHEL : `'/var/log/httpd'`
- Ubuntu : `'/var/log/apache2'`

```
node[:apache][:log_dir]
```

Attributs logrotate

Ces attributs spécifient comment faire tourner les fichiers-journaux.

delaycompress

Indique s'il faut retarder la compression d'un fichier journal fermé jusqu'au début du prochaine cycle de rotation (`'true'`/`'false'`) (chaîne). La valeur par défaut est `'true'`.

```
node[:apache][:logrotate][:delaycompress]
```

groupe

Groupe des fichiers journaux (chaîne). La valeur par défaut est `'adm'`.

```
node[:apache][:logrotate][:group]
```

mode

Mode des fichiers journaux (chaîne). La valeur par défaut est `'640'`.

```
node[:apache][:logrotate][:mode]
```

owner

Propriétaire des fichiers journaux (chaîne). La valeur par défaut est `'root'`.

```
node[:apache][:logrotate][:owner]
```

rotate

Nombre de cycles de rotation avant qu'un fichier journal fermé soit supprimé (chaîne). La valeur par défaut est `'30'`.

```
node[:apache][:logrotate][:rotate]
```

schedule

Planification de la rotation (chaîne). Les valeurs possibles sont les suivantes :

- `'daily'`
- `'weekly'`
- `'monthly'`

La valeur par défaut est `'daily'`.

```
node[:apache][:logrotate][:schedule]
```

pid_file

Fichier qui contient l'ID de processus du démon (chaîne). Les valeurs par défaut sont les suivantes :

- Amazon Linux et RHEL : `'/var/run/httpd/httpd.pid'`
- Ubuntu : `'/var/run/apache2.pid'`

```
node[:apache][:pid_file]
```

Attributs prefork

Ces attributs spécifient la configuration de pré-bifurcation.

maxclients

Nombre maximal de demandes simultanées qui seront traitées (nombre). La valeur par défaut est `400`.

> **Note**
>
> Utilisez cet attribut uniquement pour les instances qui exécutent Amazon Linux ou RHEL. Si vos instances exécutent Ubuntu 14.04 LTS, utilisez maxrequestworkers.

```
node[:apache][:prefork][:maxclients]
```

maxrequestsperchild

Nombre maximal de requêtes qu'un processus serveur enfant gère (nombre). La valeur par défaut est `10000`.

```
node[:apache][:prefork][:maxrequestsperchild]
```

maxrequestworkers

Nombre maximal de demandes simultanées qui seront traitées (nombre). La valeur par défaut est `400`.

> **Note**
>
> Utilisez cet attribut uniquement pour les instances qui exécutent Ubuntu 14.04 LTS. Si vos instances exécutent Amazon Linux ou RHEL, utilisezmaxclients .

```
node[:apache][:prefork][:maxrequestworkers]
```

maxspareservers

Nombre maximal de processus serveur inactifs (nombre). La valeur par défaut est `32`.

```
node[:apache][:prefork][:maxspareservers]
```

minspareservers

Nombre minimal de processus serveur inactifs (nombre). La valeur par défaut est `16`.

```
node[:apache][:prefork][:minspareservers]
```

serverlimit

Nombre maximal de processus qui peuvent être configurés (nombre). La valeur par défaut est `400`.

```
node[:apache][:prefork][:serverlimit]
```

startservers

Nombre de processus serveur enfant à créer au démarrage (nombre). La valeur par défaut est `16`.

```
node[:apache][:prefork][:startservers]
```

serversignature

Spécifie si et comment configurer un pied de page de fin pour les documents générés par le serveur (chaîne). Les valeurs possibles sont `'On'`, `'Off'` et `'Email'`. La valeur par défaut est `'Off'`.

```
node[:apache][:serversignature]
```

servertokens

Spécifie le type d'informations sur la version serveur incluses dans l'en-tête de réponse (chaîne) :

- `'Full'` : informations complètes. Par exemple, Serveur : Apache/2.4.2 (Unix) PHP/4.2.2 /1.2 MyMod
- `'Prod'` : nom du produit. Par exemple, serveur : Apache

- `'Major'` : version majeure. Par exemple, serveur : Apache/2
- `'Minor'` : version majeure et version mineure. Par exemple, serveur : Apache/2.4
- `'Min'` : version minimale. Par exemple, serveur : Apache/2.4.2
- `'OS'` : version avec système d'exploitation. Par exemple, serveur : Apache/2.4.2 (Unix)

La valeur par défaut est `'Prod'`.

```
node[:apache][:servertokens]
```

timeout

Durée d'attente d'Apache I/O (nombre). La valeur par défaut est `120`.

```
node[:apache][:timeout]
```

traceenable

Indique si les requêtes TRACE doivent être activées (chaîne). Les valeurs possibles sont `'On'` et `'Off'`. La valeur par défaut est `'Off'`.

```
node[:apache][:traceenable]
```

utilisateur

Nom d'utilisateur (chaîne). Les valeurs par défaut sont les suivantes :

- Amazon Linux et RHEL : `'apache'`
- Ubuntu : `'www-data'`

```
node[:apache][:user]
```

version

Version Apache (chaîne). Les valeurs par défaut sont les suivantes :

- Amazon Linux: `2.2`
- Ubuntu 14.04 LTS : `2.4`
- RHEL: `2.4`

```
node[:apache][:version]
```

Attributs worker

Ces attributs spécifient la configuration du processus de travail.

startservers

Nombre de processus serveur enfant à créer au démarrage (nombre). La valeur par défaut est `4`.

```
node[:apache][:worker][:startservers]
```

maxclients

Nombre maximal de demandes simultanées qui seront traitées (nombre). La valeur par défaut est `1024`.

```
node[:apache][:worker][:maxclients]
```

maxsparethreads

Nombre maximal de threads inactifs (nombre). La valeur par défaut est `192`.

```
node[:apache][:worker][:maxsparethreads]
```

minsparethreads

Nombre minimal de threads inactifs (nombre). La valeur par défaut est `64`.

```
node[:apache][:worker][:minsparethreads]
```

threadsperchild

Nombre de threads par processus enfant (nombre). La valeur par défaut est `64`.

```
node[:apache][:worker][:threadsperchild]
```

maxrequestsperchild

Nombre maximal de requêtes qu'un processus serveur enfant gère (nombre). La valeur par défaut est `10000`.

```
node[:apache][:worker][:maxrequestsperchild]
```

Attributs deploy

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Le fichier d'attributs `deploy.rb` du livre de recettes deploy intégré définit les attributs suivants dans l'espace de noms `opsworks`. Pour plus d'informations sur les répertoires de déploiement, consultez Recettes Deploy. Pour plus d'informations sur la façon de remplacer les attributs intégrés afin de spécifier les valeurs personnalisées, consultez Remplacement des attributs.

deploy_keep_releases

Paramètre global pour le nombre de déploiements d'applications que AWS OpsWorks Stacks stockera (nombre). La valeur par défaut est 5. Cette valeur contrôle le nombre de fois où vous pouvez annuler une application.

```
node[:opsworks][:deploy_keep_releases]
```

groupe

(Linux uniquement) Paramètre `group` du répertoire deploy de l'application (chaîne). La valeur par défaut dépend du système d'exploitation de l'instance :

- Pour les instances Ubuntu, la valeur par défaut est `www-data`.
- Pour les instances Amazon Linux ou RHEL membres d'une couche Rails App Server utilisant Nginx et Unicorn, la valeur par défaut est. `nginx`
- Pour les autres instances Amazon Linux ou RHEL, la valeur par défaut est `apache`.

```
node[:opsworks][:deploy_user][:group]
```

utilisateur

(Linux uniquement) Paramètre `user` du répertoire deploy de l'application (chaîne). La valeur par défaut est `deploy`.

```
node[:opsworks][:deploy_user][:user]
```

Attributs haproxy

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Ces attributs ne sont disponibles que sur les piles Linux.

Les `haproxy`attributs spécifient la configuration HAProxy du serveur. Pour plus d'informations, consultez HAProxyDocs. Pour plus d'informations sur la façon de remplacer les attributs intégrés afin de spécifier les valeurs personnalisées, consultez Remplacement des attributs.

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
nouvelles tentatives	server_timeout	stats_url
stats_user		

balance

Algorithme utilisé par un équilibreur de charge pour sélectionner un serveur (chaîne). La valeur par défaut est `'roundrobin'`. Les autres options sont :

- 'static-rr'
- 'leastconn'
- 'source'
- 'uri'
- 'url_param'
- 'hdr(name)'
- 'rdp-cookie'
- 'rdp-cookie(name)'

Pour plus d'informations sur ces arguments, consultez balance.

```
node[:haproxy][:balance]
```

check_interval

Intervalle de la vérification du statut (chaîne). La valeur par défaut est `'10s'`.

```
node[:haproxy][:check_interval]
```

client_timeout

Durée maximale pendant laquelle un client peut être inactif (chaîne). La valeur par défaut est `'60s'`.

```
node[:haproxy][:client_timeout]
```

connect_timeout

Durée maximale pendant laquelle HAProxy une tentative de connexion au serveur doit aboutir (chaîne). La valeur par défaut est `'10s'`.

```
node[:haproxy][:connect_timeout]
```

default_max_connections

Nombre maximal de connexions par défaut (chaîne). La valeur par défaut est `'80000'`.

```
node[:haproxy][:default_max_connections]
```

global_max_connections

Nombre maximal de connexions (chaîne). La valeur par défaut est `'80000'`.

```
node[:haproxy][:global_max_connections]
```

health_check_method

Méthode de vérification du statut (chaîne). La valeur par défaut est `'OPTIONS'`.

```
node[:haproxy][:health_check_method]
```

health_check_url

Chemin d'URL utilisé pour vérifier l'intégrité des serveurs (chaîne). La valeur par défaut est `'/'`.

```
node[:haproxy][:health_check_url ]
```

queue_timeout

Durée d'attente maximale pour une connexion gratuite (chaîne). La valeur par défaut est `'120s'`.

```
node[:haproxy][:queue_timeout]
```

http_request_timeout

Durée maximale d'attente d' HAProxy une requête HTTP complète (chaîne). La valeur par défaut est `'30s'`.

```
node[:haproxy][:http_request_timeout]
```

nouvelles tentatives

Nombre de tentatives après l'échec d'une connexion serveur (chaîne). La valeur par défaut est `'3'`.

```
node[:haproxy][:retries]
```

server_timeout

Durée maximale pendant laquelle un client peut être inactif (chaîne). La valeur par défaut est `'60s'`.

```
node[:haproxy][:server_timeout]
```

stats_url

Chemin d'URL pour la page de statistiques (chaîne). La valeur par défaut est `'/haproxy?stats'`.

```
node[:haproxy][:stats_url]
```

stats_user

Nom d'utilisateur de la page de statistiques (chaîne). La valeur par défaut est `'opsworks'`.

```
node[:haproxy][:stats_user]
```

Les `maxcon` attributs représentent un multiplicateur de facteur de charge utilisé pour calculer le nombre maximum de connexions autorisées pour HAProxy les backends. Par exemple, supposons que vous disposiez d'un serveur d'applications Rails sur une petite instance avec une `backend` valeur de 4, ce qui signifie que AWS OpsWorks Stacks configurera quatre processus Rails pour cette instance. Si vous utilisez la `maxcon_factor_rails_app` valeur par défaut de 7, 28 (4*7) connexions au serveur Rails HAProxy seront gérées.

maxcon_factor_nodejs_app

Facteur maxcon pour un serveur d'applications Node.js (nombre). La valeur par défaut est `10`.

```
node[:haproxy][:maxcon_factor_nodejs_app]
```

maxcon_factor_nodejs_app_ssl

Facteur maxcon pour un serveur d'applications Node.js avec SSL (nombre). La valeur par défaut est `10`.

```
node[:haproxy][:maxcon_factor_nodejs_app_ssl]
```

maxcon_factor_php_app

Facteur maxcon pour un serveur d'applications PHP (nombre). La valeur par défaut est `10`.

```
node[:haproxy][:maxcon_factor_php_app]
```

maxcon_factor_php_app_ssl

Facteur maxcon pour un serveur d'applications PHP avec SSL (nombre). La valeur par défaut est `10`.

```
node[:haproxy][:maxcon_factor_php_app_ssl]
```

maxcon_factor_rails_app

Facteur maxcon pour un serveur d'applications Rails (nombre). La valeur par défaut est `7`.

```
node[:haproxy][:maxcon_factor_rails_app]
```

maxcon_factor_rails_app_ssl

Facteur maxcon pour un serveur d'applications Rails avec SSL (nombre). La valeur par défaut est `7`.

```
node[:haproxy][:maxcon_factor_rails_app_ssl]
```

maxcon_factor_static

Facteur maxcon pour un serveur web statique (nombre). La valeur par défaut est `15`.

```
node[:haproxy][:maxcon_factor_static]
```

maxcon_factor_static_ssl

Facteur maxcon pour un serveur web statique avec SSL (nombre). La valeur par défaut est 15.

```
node[:haproxy][:maxcon_factor_static_ssl]
```

Attributs memcached

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Ces attributs ne sont disponibles que sur les piles Linux.

Les attributs `memcached` spécifient la configuration du serveur Memcached. Pour plus d'informations sur la façon de remplacer les attributs intégrés afin de spécifier les valeurs personnalisées, consultez Remplacement des attributs.

memory	max_connections	pid_file
port	start_command	stop_command
utilisateur		

memory

Mémoire maximale à utiliser, en Mo (nombre). La valeur par défaut est 512.

```
node[:memcached][:memory]
```

max_connections

Nombre maximal de connexions (chaîne). La valeur par défaut est `'4096'`.

```
node[:memcached][:max_connections]
```

pid_file

Fichier qui contient l'ID de processus du démon (chaîne). La valeur par défaut est `'var/run/memcached.pid'`.

```
node[:memcached][:pid_file]
```

port

Port sur lequel écouter (nombre). La valeur par défaut est `11211`.

```
node[:memcached][:port]
```

start_command

Commande start (chaîne). La valeur par défaut est `'/etc/init.d/memcached start'`.

```
node[:memcached][:start_command]
```

stop_command

Commande stop (chaîne). La valeur par défaut est `'/etc/init.d/memcached stop'`.

```
node[:memcached][:stop_command]
```

utilisateur

Utilisateur (chaîne). La valeur par défaut est `'nobody'`.

```
node[:memcached][:user]
```

Attributs mysql

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Ces attributs ne sont disponibles que sur les piles Linux.

Les attributs `mysql` spécifient la configuration principale MySQL. Pour plus d'informations, consultez Variables système serveur. Pour plus d'informations sur la façon de remplacer les attributs intégrés afin de spécifier les valeurs personnalisées, consultez Remplacement des attributs.

basedir	bind_address	clients
conf_dir	confd_dir	datadir
grants_path	mysql_bin	mysqladmin_bin
pid_file	port	root_group
server_root_password	socket	Attributs tunable

basedir

Répertoire de base (chaîne). La valeur par défaut est `'/usr'`.

```
node[:mysql][:basedir]
```

bind_address

Adresse sur laquelle MySQL écoute (chaîne). La valeur par défaut est `'0.0.0.0'`.

```
node[:mysql][:bind_address]
```

clients

Liste de clients (liste de chaînes).

```
node[:mysql][:clients]
```

conf_dir

Répertoire qui contient le fichier de configuration (chaîne). Les valeurs par défaut sont les suivantes :

- Amazon Linux et RHEL : `'/etc'`
- Ubuntu : `'/etc/mysql'`

```
node[:mysql][:conf_dir]
```

confd_dir

Répertoire qui contient les fichiers de configuration supplémentaires (chaîne). La valeur par défaut est `'/etc/mysql/conf.d'`.

```
node[:mysql][:confd_dir]
```

datadir

Répertoire des données (chaîne). La valeur par défaut est `'/var/lib/mysql'`.

```
node[:mysql][:datadir]
```

grants_path

Emplacement de la table d'attribution (chaîne). La valeur par défaut est `'/etc/mysql_grants.sql'`.

```
node[:mysql][:grants_path]
```

mysql_bin

Emplacement des fichiers binaires mysql (chaîne). La valeur par défaut est `'/usr/bin/mysql'`.

```
node[:mysql][:mysql_bin]
```

mysqladmin_bin

Emplacement mysqladmin (chaîne). La valeur par défaut est `'/usr/bin/mysqladmin'`.

```
node[:mysql][:mysqladmin_bin]
```

pid_file

Fichier qui contient l'ID de processus du démon (chaîne). La valeur par défaut est `'/var/run/mysqld/mysqld.pid'`.

```
node[:mysql][:pid_file]
```

port

Port sur lequel le serveur écoute (nombre). La valeur par défaut est `3306`.

```
node[:mysql][:port]
```

root_group

Groupe racine (chaîne). La valeur par défaut est `'root'`.

```
node[:mysql][:root_group]
```

server_root_password

Mot de passe racine du serveur (chaîne). La valeur par défaut est générée de façon aléatoire.

```
node[:mysql][:server_root_password]
```

socket

Emplacement du fichier socket (chaîne). La valeur par défaut est `'/var/lib/mysql/mysql.sock'`. Les valeurs par défaut sont les suivantes :

- Amazon Linux et RHEL : `'/var/lib/mysql/mysql.sock'`
- Ubuntu : `'/var/run/mysqld/mysqld.sock'`

```
node[:mysql][:socket]
```

Attributs tunable

Les attributs tunable sont utilisés pour le réglage des performances.

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

Nombre maximal de demandes en attente (chaîne). La valeur par défaut est `'128'`.

```
node[:mysql][:tunable][:back_log]
```

innodb_additional_mem_pool_size

Taille du pool qu'Innodb utilise pour stocker les structures de données internes (chaîne). La valeur par défaut est `'20M'`.

```
node[:mysql][:tunable][:innodb_additional_mem_pool_size]
```

innodb_buffer_pool_size

Taille du pool de mémoires tampons Innodb (chaîne). La valeur de l'attribut est définie par AWS OpsWorks Stacks et dépend du type d'instance, mais vous pouvez la remplacer en utilisant du JSON personnalisé ou un fichier d'attribut personnalisé.

```
node[:mysql][:tunable][:innodb_buffer_pool_size]
```

innodb_flush_log_at_trx_commit

Fréquence à laquelle Innodb vide la mémoire tampon du journal (chaîne). La valeur par défaut est `'2'`. Pour plus d'informations, consultez innodb_flush_log_at_trx_commit.

```
node[:mysql][:tunable][:innodb_flush_log_at_trx_commit]
```

innodb_lock_wait_timeout

Durée maximale, en secondes, pendant laquelle une transaction Innodb attend un verrou de ligne (chaîne). La valeur par défaut est `'50'`.

```
node[:mysql][:tunable][:innodb_lock_wait_timeout]
```

key_buffer

Taille du tampon d'index (chaîne). La valeur par défaut est `'250M'`.

```
node[:mysql][:tunable][:key_buffer]
```

log_slow_queries

Emplacement du fichier journal des requêtes lentes (chaîne). La valeur par défaut est `'/var/log/mysql/mysql-slow.log'`.

```
node[:mysql][:tunable][:log_slow_queries]
```

long_query_time

Durée, en secondes, nécessaire pour définir une requête comme requête longue (chaîne). La valeur par défaut est `'1'`.

```
node[:mysql][:tunable][:long_query_time]
```

max_allowed_packet

Taille de paquet maximale autorisée (chaîne). La valeur par défaut est `'32M'`.

```
node[:mysql][:tunable][:max_allowed_packet]
```

max_connections

Nombre maximal de connexions client simultanées (chaîne). La valeur par défaut est `'2048'`.

```
node[:mysql][:tunable][:max_connections]
```

max_heap_table_size

Taille maximale des tables `MEMORY` créées par l'utilisateur (chaîne). La valeur par défaut est `'32M'`.

```
node[:mysql][:tunable][:max_heap_table_size]
```

net_read_timeout

Durée d'attente, en secondes, de données supplémentaires à partir d'une connexion (chaîne). La valeur par défaut est `'30'`.

```
node[:mysql][:tunable][:net_read_timeout]
```

net_write_timeout

Durée d'attente, en secondes, pour qu'un bloc soit écrit sur une connexion (chaîne). La valeur par défaut est `'30'`.

```
node[:mysql][:tunable][:net_write_timeout]
```

query_cache_limit

Taille maximale d'une requête mise en cache (chaîne). La valeur par défaut est `'2M'`.

```
node[:mysql][:tunable][:query_cache_limit]
```

query_cache_size

Taille du cache de requête (chaîne). La valeur par défaut est `'128M'`.

```
node[:mysql][:tunable][:query_cache_size]
```

query_cache_type

Type du cache de requête (chaîne). Les valeurs possibles sont les suivantes :

- `'0'` : aucune mise en cache ou récupération des données mises en cache.
- `'1'` : instructions de mise en cache qui ne commencent pas par `SELECT SQL_NO_CACHE`.
- `'2'` : instructions de mise en cache qui commencent par `SELECT SQL_CACHE`.

La valeur par défaut est `'1'`.

```
node[:mysql][:tunable][:query_cache_type]
```

thread_cache_size

Nombre de threads client mis en cache pour la réutilisation (chaîne). La valeur par défaut est `'8'`.

```
node[:mysql][:tunable][:thread_cache_size]
```

thread_stack

Taille de pile de chaque thread (chaîne). La valeur par défaut est `'192K'`.

```
node[:mysql][:tunable][:thread_stack]
```

wait_timeout

Durée d'attente, en secondes, sur une connexion non interactive. La valeur par défaut est `'180'` (chaîne).

```
node[:mysql][:tunable][:wait_timeout]
```

table_cache

Nombre de tables ouvertes (chaîne). La valeur par défaut est `'2048'`.

```
node[:mysql][:tunable][:table_cache]
```

Attributs nginx

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement

aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Ces attributs ne sont disponibles que sur les piles Linux.

Les attributs `nginx` spécifient la configuration Nginx. Pour plus d'informations, consultez Index des directives. Pour plus d'informations sur la façon de remplacer les attributs intégrés afin de spécifier les valeurs personnalisées, consultez Remplacement des attributs.

binary	dir	gzip
gzip_comp_level	gzip_disable	gzip_http_version
gzip_proxied	gzip_static	gzip_types
gzip_vary	keepalive	keepalive_timeout
log_dir	utilisateur	server_names_hash_bucket_size
worker_processes	worker_connections	

binary

Emplacement des fichiers binaires Nginx (chaîne). La valeur par défaut est `'/usr/sbin/nginx'`.

```
node[:nginx][:binary]
```

dir

Emplacement de fichiers tels que les fichiers de configuration (chaîne). La valeur par défaut est `'/etc/nginx'`.

```
node[:nginx][:dir]
```

gzip

Indique si la compression gzip est activée (chaîne). Les valeurs possibles sont `'on'` et `'off'`. La valeur par défaut est `'on'`.

> **Warning**
>
> La compression peut entraîner des risques de sécurité. Pour désactiver complètement la compression, définissez l'attribut comme suit :
>
> ```
> node[:nginx][:gzip] = 'off'
> ```

```
node[:nginx][:gzip]
```

gzip_comp_level

Le niveau de compression, qui peut aller de 1 à 9, 1 correspondant à la compression la plus faible (chaîne). La valeur par défaut est `'2'`.

```
node[:nginx][:gzip_comp_level]
```

gzip_disable

Désactive la compression gzip pour les agents utilisateur spécifiés (chaîne). La valeur est une expression régulière et la valeur par défaut est `'MSIE [1-6].(?!.*SV1)'`.

```
node[:nginx][:gzip_disable]
```

gzip_http_version

Active la compression gzip pour une version HTTP spécifiée (chaîne). La valeur par défaut est `'1.0'`.

```
node[:nginx][:gzip_http_version]
```

gzip_proxied

Indique si et comment compresser la réponse aux requêtes proxy, ce qui peut prendre l'une des valeurs suivantes (chaîne) :

- `'off'` : ne pas compresser les demandes en proxy
- `'expired'` : compresser si l'en-tête Expire empêche la mise en cache
- `'no-cache'` : compresser si l'en-tête Cache-Control est défini sur « no-cache »
- `'no-store'` : compresser si l'en-tête Cache-Control est défini sur « no-store »
- `'private'` : compresser si l'en-tête Cache-Control est défini sur « private »
- `'no_last_modified'` : compresser si Last-Modified n'est pas défini
- `'no_etag'`: compresser si la requête n'a pas d' ETagen-tête
- `'auth'` : compresser si la demande inclut un en-tête Authorization
- `'any'` : compresser toutes les demandes en proxy

La valeur par défaut est `'any'`.

```
node[:nginx][:gzip_proxied]
```

gzip_static

Indique si le module statique gzip est activé (chaîne). Les valeurs possibles sont `'on'` et `'off'`. La valeur par défaut est `'on'`.

```
node[:nginx][:gzip_static]
```

gzip_types

Liste des types MIME à compresser (liste de chaînes). La valeur par défaut est `['text/plain', 'text/html', 'text/css', 'application/x-javascript', 'text/xml', 'application/xml', 'application/xml+rss', 'text/javascript']`.

```
node[:nginx][:gzip_types]
```

gzip_vary

Indique si vous voulez activer un en-tête de réponse `Vary:Accept-Encoding` (chaîne). Les valeurs possibles sont `'on'` et `'off'`. La valeur par défaut est `'on'`.

```
node[:nginx][:gzip_vary]
```

keepalive

Indique si vous voulez qu'une connexion demeure toujours active (chaîne). Les valeurs possibles sont `'on'` et `'off'`. La valeur par défaut est `'on'`.

```
node[:nginx][:keepalive]
```

keepalive_timeout

Durée maximale, en secondes, pendant laquelle une connexion toujours active demeure ouverte (nombre). La valeur par défaut est `65`.

```
node[:nginx][:keepalive_timeout]
```

log_dir

Emplacement des fichiers journaux (chaîne). La valeur par défaut est `'/var/log/nginx'`.

```
node[:nginx][:log_dir]
```

utilisateur

Utilisateur (chaîne). Les valeurs par défaut sont les suivantes :

- Amazon Linux et RHEL : `'www-data'`
- Ubuntu : `'nginx'`

```
node[:nginx][:user]
```

server_names_hash_bucket_size

Taille du compartiment pour les tables de hachage des noms de serveur, qui peut être définie sur `32`, `64` ou `128` (nombre). La valeur par défaut est `64`.

```
node[:nginx][:server_names_hash_bucket_size]
```

worker_processes

Nombre de processus de travail (nombre). La valeur par défaut est `10`.

```
node[:nginx][:worker_processes]
```

worker_connections

Nombre maximal de connexions de travail (nombre). La valeur par défaut est `1024`. Le nombre maximal de clients est défini sur `worker_processes * worker_connections`.

```
node[:nginx][:worker_connections]
```

Attributs opsworks_berkshelf

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Ces attributs ne sont disponibles que sur les piles Linux.

Les attributs `opsworks_berkshelf` spécifient la configuration Berkshelf. Pour plus d'informations, consultez Berkshelf. Pour plus d'informations sur la façon de remplacer les attributs intégrés afin de spécifier les valeurs personnalisées, consultez Remplacement des attributs.

debug

Indique si vous souhaitez inclure les informations de débogage Berkshelf dans le journal Chef (valeur booléenne). La valeur par défaut est `false`.

```
node['opsworks_berkshelf]['debug']
```

Attributs opsworks_java

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Ces attributs ne sont disponibles que sur les piles Linux.

Les attributs `opsworks_java` spécifient la configuration du serveur Tomcat. Pour plus d'informations, consultez Référence de configuration Apache Tomcat. Pour plus d'informations sur la façon de remplacer les attributs intégrés afin de spécifier les valeurs personnalisées, consultez Remplacement des attributs.

datasources	java_app_server_version	java_shared_lib_dir
Attributs jvm_pkg	custom_pkg_location_url_debian	java_home_basedir
custom_pkg_location_url_rhel	use_custom_pkg_location	jvm_options
jvm_version	Attributs tomcat	

datasources

Ensemble d'attributs qui définissent les noms de ressource JNDI (chaîne). Pour plus d'informations sur l'utilisation de cet attribut, consultez Déploiement d'une application JSP avec une base de données principale. La valeur par défaut est un hachage vide, qui peut être rempli avec des mappages personnalisés entre les noms courts d'application et les noms JNDI. Pour de plus amples informations, veuillez consulter Déploiement d'une application JSP avec une base de données principale.

```
node['opsworks_java']['datasources']
```

java_app_server_version

Version du serveur d'applications Java (nombre). La valeur par défaut est 7. Vous pouvez remplacer cet attribut pour spécifier la version 6. Si vous installez un JDK personnalisé, cet attribut est ignoré.

```
node['opsworks_java']['java_app_server_version']
```

java_shared_lib_dir

Répertoire des bibliothèques Java partagées (chaîne). La valeur par défaut est `/usr/share/java`.

```
node['opsworks_java']['java_shared_lib_dir']
```

Attributs jvm_pkg

Ensemble d'attributs que vous pouvez remplacer pour installer un JDK personnalisé.

use_custom_pkg_location

Indique s'il convient d'installer un JDK personnalisé au lieu d'OpenJDK (valeur booléenne). La valeur par défaut est `false`.

```
node['opsworks_java']['jvm_pkg']['use_custom_pkg_location']
```

custom_pkg_location_url_debian

Emplacement du package JDK à installer sur les instances Ubuntu (chaîne). La valeur par défaut est `'http://aws.amazon.com/'`, qui est simplement une valeur d'initialisation sans signification propre. Si vous souhaitez installer un JDK personnalisé, vous devez remplacer cet attribut et le définir avec l'URL appropriée.

```
node['opsworks_java']['jvm_pkg']['custom_pkg_location_url_debian']
```

custom_pkg_location_url_rhel

Emplacement du package JDK à installer sur les instances Amazon Linux et RHEL (chaîne). La valeur par défaut est `'http://aws.amazon.com/'`, qui est simplement une valeur

d'initialisation sans signification propre. Si vous souhaitez installer un JDK personnalisé, vous devez remplacer cet attribut et le définir avec l'URL appropriée.

```
node['opsworks_java']['jvm_pkg']['custom_pkg_location_url_rhel']
```

java_home_basedir

Répertoire sur lequel le package JDK sera extrait (chaîne). La valeur par défaut est `/usr/local`. Vous n'avez pas besoin de spécifier ce paramètre pour les packages RPM ; ils incluent une structure de répertoire complète.

```
node['opsworks_java']['jvm_pkg']['java_home_basedir']
```

jvm_options

Options de ligne de commande de machine virtuelle Java, qui vous permettent de spécifier des paramètres tels que la taille du tas (chaîne). Un ensemble courant d'options est `-Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC`. La valeur par défaut est aucune option.

```
node['opsworks_java']['jvm_options']
```

jvm_version

Version OpenJDK (nombre). La valeur par défaut est 7. Vous pouvez remplacer cet attribut pour spécifier OpenJDK version 6. Si vous installez un JDK personnalisé, cet attribut est ignoré.

```
node['opsworks_java']['jvm_version']
```

Attributs tomcat

Ensemble d'attributs que vous pouvez remplacer pour installer la configuration Tomcat par défaut.

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

Port AJP (nombre). La valeur par défaut est `8009`.

```
node['opsworks_java']['tomcat]['ajp_port']
```

apache_tomcat_bind_mod

Module proxy (chaîne). La valeur par défaut est `proxy_http`. Vous pouvez remplacer cet attribut pour spécifier le module proxy AJP, `proxy_ajp`.

```
node['opsworks_java']['tomcat]['apache_tomcat_bind_mod']
```

apache_tomcat_bind_path

Chemin de liaison Apache-Tomcat (chaîne). La valeur par défaut est `/`. Vous ne devez pas remplacer cet attribut ; la modification du chemin de liaison peut entraîner l'arrêt de l'application.

```
node['opsworks_java']['tomcat]['apache_tomcat_bind_path']
```

auto_deploy

Indique si le déploiement automatique est autorisé (valeur booléenne). La valeur par défaut est `true`.

```
node['opsworks_java']['tomcat]['auto_deploy']
```

connection_timeout

Délai de connexion, en millisecondes (nombre). La valeur par défaut est `20000` (20 secondes).

```
node['opsworks_java']['tomcat]['connection_timeout']
```

mysql_connector_jar

Fichier JAR de la bibliothèque du connecteur MySQL (chaîne). La valeur par défaut est `mysql-connector-java.jar`.

```
node['opsworks_java']['tomcat]['mysql_connector_jar']
```

port

Port standard (nombre). La valeur par défaut est `8080`.

```
node['opsworks_java']['tomcat]['port']
```

secure_port

Port sécurisé (nombre). La valeur par défaut est `8443`.

```
node['opsworks_java']['tomcat]['secure_port']
```

shutdown_port

Port de fermeture (nombre). La valeur par défaut est `8005`.

```
node['opsworks_java']['tomcat]['shutdown_port']
```

threadpool_max_threads

Nombre maximal de threads dans le pool (nombre). La valeur par défaut est `150`.

```
node['opsworks_java']['tomcat]['threadpool_max_threads']
```

threadpool_min_spare_threads

Nombre minimal de threads de secours dans le pool (nombre). La valeur par défaut est `4`.

```
node['opsworks_java']['tomcat]['threadpool_min_spare_threads']
```

unpack_wars

Indique s'il faut décompresser les fichiers WAR (valeur booléenne). La valeur par défaut est `true`.

```
node['opsworks_java']['tomcat]['unpack_wars']
```

uri_encoding

Encodage de l'URI (chaîne). La valeur par défaut est `UTF-8`.

```
node['opsworks_java']['tomcat]['uri_encoding']
```

use_ssl_connector

Indique s'il faut utiliser un connecteur SSL (valeur booléenne). La valeur par défaut est `false`.

```
node['opsworks_java']['tomcat]['use_ssl_connector']
```

use_threadpool

Indique s'il faut utiliser un pool de threads (valeur booléenne). La valeur par défaut est `false`.

```
node['opsworks_java']['tomcat]['use_threadpool']
```

userdatabase_pathname

Nom du chemin de la base de données utilisateur (chaîne). La valeur par défaut est `conf/tomcat-users.xml`.

```
node['opsworks_java']['tomcat]['userdatabase_pathname']
```

Attributs passenger_apache2

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Ces attributs ne sont disponibles que sur les piles Linux.

Les attributs `passenger_apache2` spécifient la configuration Phusion Passenger. Pour plus d'informations, consultez Guide de l'utilisateur Phusion Passenger, version Apache. Pour plus d'informations sur la façon de remplacer les attributs intégrés afin de spécifier les valeurs personnalisées, consultez Remplacement des attributs.

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

Indique s'il faut afficher une page d'erreur conviviale quand une application ne peut pas démarrer (chaîne). Cet attribut peut être défini sur « on » ou « off » ; la valeur par défaut est « off ».

```
node[:passenger][:friendly_error_pages]
```

gem_bin

Emplacement des fichiers binaires Gem (chaîne). La valeur par défaut est `'/usr/local/bin/gem'`.

```
node[:passenger][:gem_bin]
```

gems_path

Chemin d'accès GEM (chaîne). La valeur par défaut dépend de la version Ruby. Par exemple :

- Version Ruby 1.8 : `'/usr/local/lib/ruby/gems/1.8/gems'`
- Version Ruby 1.9 : `'/usr/local/lib/ruby/gems/1.9.1/gems'`

```
node[:passenger][:gems_path]
```

high_performance_mode

Indique s'il faut utiliser le mode hautes performances de Passenger (chaîne). Les valeurs possibles sont `'on'` et `'off'`. La valeur par défaut est `'off'`.

```
node[:passenger][:high_performance_mode ]
```

root_path

Répertoire racine de Passenger (chaîne). La valeur par défaut dépend des versions Ruby et Passenger. Dans la syntaxe Chef, la valeur est `"#{node[:passenger][:gems_path]}/passenger-#{passenger[:version]}"`.

```
node[:passenger][:root_path]
```

max_instances_per_app

Nombre maximal de processus d'application par application (nombre). La valeur par défaut est `0`. Pour de plus amples informations, veuillez consulter PassengerMaxInstancesPerApp.

```
node[:passenger][:max_instances_per_app]
```

max_pool_size

Nombre maximal de processeurs d'application (nombre). La valeur par défaut est 8. Pour de plus amples informations, veuillez consulter PassengerMaxPoolSize.

```
node[:passenger][:max_pool_size]
```

max_requests

Nombre maximal de requêtes (nombre). La valeur par défaut est `0`.

```
node[:passenger][:max_requests]
```

module_path

Chemin du module (chaîne). Les valeurs par défaut sont les suivantes :

- Amazon Linux et RHEL : `"#{node['apache']['libexecdir']}/mod_passenger.so"`
- Ubuntu: `"#{passenger[:root_path]}/ext/apache2/mod_passenger.so"`

```
node[:passenger][:module_path]
```

pool_idle_time

Durée maximale, en secondes, pendant laquelle un processus d'application peut être inactif (nombre). La valeur par défaut est `14400` (4 heures). Pour de plus amples informations, veuillez consulter PassengerPoolIdleTime.

```
node[:passenger][:pool_idle_time]
```

rails_app_spawner_idle_time

Durée maximale d'inactivité pour le générateur d'application Rails (nombre). Si l'attribut a la valeur zéro, le générateur d'application n'expire pas. La valeur par défaut est `0`. Pour plus d'informations, consultez Présentation des méthodes de génération.

```
node[:passenger][:rails_app_spawner_idle_time]
```

rails_framework_spawner_idle_time

Durée maximale d'inactivité pour le générateur d'infrastructure Rails (nombre). Si l'attribut a la valeur zéro, le générateur d'infrastructure n'expire pas. La valeur par défaut est `0`. Pour plus d'informations, consultez Présentation des méthodes de génération.

```
node[:passenger][:rails_framework_spawner_idle_time]
```

rails_spawn_method

Méthode de génération Rails (chaîne). La valeur par défaut est `'smart-lv2'`. Pour plus d'informations, consultez Présentation des méthodes de génération.

```
node[:passenger][:rails_spawn_method]
```

ruby_bin

Emplacement des fichiers binaires Ruby (chaîne). La valeur par défaut est `'/usr/local/bin/ruby'`.

```
node[:passenger][:ruby_bin]
```

ruby_wrapper_bin

Emplacement du script du wrapper Ruby (chaîne). La valeur par défaut est `'/usr/local/bin/ruby_gc_wrapper.sh'`.

```
node[:passenger][:ruby_wrapper_bin]
```

stat_throttle_rate

Vitesse à laquelle Passenger effectue les contrôles du système de fichiers (nombre). La valeur par défaut est 5, ce qui signifie que les contrôles sont réalisés toutes les 5 secondes au plus. Pour de plus amples informations, veuillez consulter PassengerStatThrottleRate .

```
node[:passenger][:stat_throttle_rate]
```

version

Version (chaîne). La valeur par défaut est `'3.0.9'`.

```
node[:passenger][:version]
```

Attributs ruby

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Ces attributs ne sont disponibles que sur les piles Linux.

Les attributs `ruby` spécifient la version Ruby utilisée par les applications. Notez que l'utilisation de l'attribut a changé avec l'introduction de la gestion sémantique des versions dans Ruby 2.1. Pour plus d'informations sur la façon de spécifier une version, exemples inclus, consultez Versions de Ruby. Pour plus de détails sur la façon dont AWS OpsWorks Stacks détermine la version de Ruby, consultez le fichier d'attributs intégré, ruby.rb. Pour plus d'informations sur la façon de remplacer les attributs intégrés afin de spécifier les valeurs personnalisées, consultez Remplacement des attributs.

full_version

Numéro de version complet (chaîne). Vous ne devez pas remplacer cet attribut. Utilisez à la place [:opsworks][:ruby_version] et l'attribut de version du correctif approprié pour spécifier une version.

```
[:ruby][:full_version]
```

major_version

Numéro de version majeure (chaîne). Vous ne devez pas remplacer cet attribut. Utilisez à la place [:opsworks][:ruby_version] pour spécifier la version majeure.

```
[:ruby][:major_version]
```

minor_version

Numéro de version mineure (chaîne). Vous ne devez pas remplacer cet attribut. Utilisez à la place [:opsworks][:ruby_version] pour spécifier la version mineure.

```
[:ruby][:minor_version]
```

patch

Niveau de correctif (chaîne). L'attribut est valide pour Ruby 2.0.0 et version antérieure. Pour les versions Ruby postérieures, utilisez l'attribut `patch_version`.

```
[:ruby][:patch]
```

Le numéro de patch doit être précédé de p. Par exemple, vous utiliserez le JSON personnalisé suivant pour spécifier le niveau de correctif 484.

```
{
  "ruby":{"patch":"p484"}
}
```

patch_version

Numéro du correctif (chaîne). L'attribut est valide pour Ruby 2.1 et version ultérieure. Pour les versions Ruby antérieures, utilisez l'attribut `patch`.

```
[:ruby][:patch_version]
```

pkgrelease

Numéro de version du package (chaîne).

```
[:ruby][:pkgrelease]
```

Attributs unicorn

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Ces attributs ne sont disponibles que sur les piles Linux.

Les attributs `unicorn` spécifient la configuration Unicorn. Pour plus d'informations, consultez Unicorn::Configurator. Pour plus d'informations sur la façon de remplacer les attributs intégrés afin de spécifier les valeurs personnalisées, consultez Remplacement des attributs.

accept_filter	backlog	delay
tcp_nodelay	tcp_nopush	preload_app
timeout	tries	version
worker_processes		

accept_filter

Le filtre d'acceptation, `'httpready'` ou `'dataready'` (chaîne). La valeur par défaut est `'httpready'`.

```
node[:unicorn][:accept_filter]
```

backlog

Nombre maximal de requêtes que la file d'attente peut contenir (nombre). La valeur par défaut est `1024`.

```
node[:unicorn][:backlog]
```

delay

Durée d'attente, en secondes, d'une nouvelle tentative de liaison d'un socket (nombre). La valeur par défaut est `0.5`.

```
node[:unicorn][:delay]
```

preload_app

Indique s'il faut précharger une application avant de dupliquer un processus de travail (valeur booléenne). La valeur par défaut est `true`.

```
node[:unicorn][:preload_app]
```

tcp_nodelay

Indique s'il faut désactiver l'algorithme de Nagle pour les sockets TCP (valeur booléenne). La valeur par défaut est `true`.

```
node[:unicorn][:tcp_nodelay]
```

tcp_nopush

Indique s'il faut activer TCP_CORK (valeur booléenne). La valeur par défaut est `false`.

```
node[:unicorn][:tcp_nopush]
```

timeout

Durée maximale, en secondes, qu'un travail est autorisé à utiliser pour chaque demande (nombre). Les travaux qui dépassent la valeur de délai d'attente sont terminés. La valeur par défaut est `60`.

```
node[:unicorn][:timeout]
```

tries

Nombre maximal de tentatives de liaison à un socket (nombre). La valeur par défaut est `5`.

```
node[:unicorn][:tries]
```

version

Version Unicorn (chaîne). La valeur par défaut est `'4.7.0'`.

```
node[:unicorn][:version]
```

worker_processes

Nombre de processus de travail (nombre). La valeur par défaut est `max_pool_size`, s'il existe, et `4` dans le cas contraire.

```
node[:unicorn][:worker_processes]
```

Résolution des problèmes de Chef 11.10 et versions antérieures pour Linux

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Note

Pour obtenir des informations de dépannage supplémentaires, consultez Guide de débogage et dépannage.

Journaux Chef pour Chef 11.10 et versions antérieures pour Linux

AWS OpsWorks Stacks stocke les journaux Chef de chaque instance dans son `/var/lib/aws/opsworks/chef` répertoire. Vous avez besoin de privilèges sudo pour accéder à ce répertoire. Le journal pour chaque exécution se trouve dans un fichier nommé *YYYY*`-`*MM*`-`*DD*`-`*HH*`-`*MM*`-`*SS*`-`*NN*`.log`.

Pour plus d'informations, consultez les ressources suivantes :

- Affichage d'un journal de Chef avec la console
- Affichage d'un journal de Chef avec l'API ou l'interface de ligne de commande
- Interprétation d'un journal Chef
- Erreurs courantes dans le journal de Chef

Utilisation de AWS OpsWorks Stacks avec d'autres services AWS

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez faire en sorte que les serveurs d'applications exécutés dans une pile AWS OpsWorks Stacks utilisent divers services AWS qui ne sont pas directement intégrés à AWS OpsWorks Stacks. Par exemple, vos serveurs d'applications peuvent utiliser Amazon RDS comme base de données principale. Vous pouvez accéder à ces services en utilisant le modèle général suivant :

1. Créez et configurez le service AWS à l'aide de la console AWS, de l'API ou de l'interface de ligne de commande, puis enregistrez les données de configuration nécessaires dont l'application a besoin pour accéder au service, comme le nom d'hôte ou le port.
2. Créez une ou plusieurs recettes personnalisées pour configurer l'application de telle sorte qu'elle accède au service.

   La recette obtient les données de configuration à partir des attributs JSON de configuration et de déploiement de pile que vous définissez avec le JSON personnalisé avant d'exécuter les recettes.
3. Attribuez la recette personnalisée à l'événement du cycle de vie Deploy de la couche serveur d'applications.
4. Créez un objet JSON personnalisé qui attribue les valeurs appropriées aux attributs des données de configuration, et ajoutez-le à votre JSON de configuration et de déploiement de pile.
5. Déployez l'application sur la pile.

Le déploiement exécute les recettes personnalisées, qui utilisent les valeurs des données de configuration que vous avez définies dans le JSON personnalisé pour configurer l'application afin qu'elle puisse accéder au service.

Cette section explique comment permettre aux serveurs d'applications AWS OpsWorks Stacks d'accéder à divers services AWS. Il part du principe que vous êtes déjà familiarisé avec les livres de recettes Chef et avec la façon dont les recettes peuvent utiliser les attributs JSON de configuration et de déploiement de pile pour configurer les applications, généralement en créant des fichiers de configuration. Dans le cas contraire, lisez d'abord Livres de recettes et recettes et Personnalisation des piles AWS OpsWorks.

Rubriques

- Utilisation d'un magasin de données principal
- Utiliser ElastiCache Redis comme magasin de valeurs clés en mémoire
- Utilisation d'un compartiment Amazon S3
- Utilisation AWS CodePipeline avec AWS OpsWorks Stacks

Utilisation d'un magasin de données principal

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les piles de serveurs d'applications incluent généralement un serveur de base de données qui fournit un magasin de données principal. AWS OpsWorks Stacks fournit un support intégré pour les serveurs MySQL via la couche MySQL et pour plusieurs types de serveurs de base de données via la couche Amazon Relational Database Service (Amazon RDS). Cependant, vous pouvez facilement personnaliser une pile pour que les serveurs d'applications utilisent d'autres serveurs de base de données tels qu'Amazon DynamoDB ou MongoDB. Cette rubrique décrit la procédure de base pour connecter un serveur d'applications à un serveur de base de données AWS. Elle utilise

la pile et l'application de Mise en route des piles Linux Chef 11 pour montrer comment connecter manuellement un serveur d'applications PHP à une base de données RDS. Bien que l'exemple soit basé sur une pile Linux, les principes de base s'appliquent aussi aux piles Windows. Pour un exemple d'incorporation d'un serveur de base de données MongoDB dans une pile, consultez Déploiement de MongoDB avec. OpsWorks

> **Note**
>
> Cette rubrique utilise Amazon RDS comme exemple pratique. Toutefois, si vous souhaitez utiliser une base de données Amazon RDS avec votre stack, il est beaucoup plus facile d'utiliser une couche Amazon RDS.

Rubriques

- Comment configurer une connexion de base de données
- Comment connecter une instance de serveur d'applications à Amazon RDS

Comment configurer une connexion de base de données

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous configurez la connexion entre un serveur d'applications et de sa base de données principale à l'aide d'une recette personnalisée. La recette configure le serveur d'applications comme requis, généralement en créant un fichier de configuration. La recette obtient les données de connexion telles que le nom de l'hôte et de la base de données à partir d'un ensemble d'attributs de configuration de la pile et d'attributs de déploiement que AWS OpsWorks Stacks installe sur chaque instance.

Par exemple, l'étape 2 de Mise en route des piles Linux Chef 11 est basée sur une pile nommée MyStack avec deux couches, PHP App Server et MySQL, chacune avec une instance. Vous

déployez une application nommée Simple PHPApp sur l'instance PHP App Server qui utilise la base de données de l'instance MySQL comme magasin de données principal. Lorsque vous déployez l'application, AWS OpsWorks Stacks installe les attributs de configuration et de déploiement de pile qui contiennent les informations de connexion de base de données. L'exemple suivant montre les attributs de connexion de base de données, représentés sous la forme JSON :

```
{
  ...
  "deploy": {
    "simplephpapp": {
      ...
      "database": {
        "reconnect": true,
        "password": null,
        "username": "root",
        "host": null,
        "database": "simplephpapp"
        ...
      },
      ...
    }
  }
}
```

Les valeurs d'attribut sont fournies par AWS OpsWorks Stacks et sont générées ou basées sur les informations fournies par l'utilisateur.

Pour permettre PHPApp à Simple d'accéder au magasin de données, vous devez configurer la connexion entre le serveur d'applications PHP et la base de données MySQL en attribuant une recette personnalisée nommée `appsetup.rb` à l'événement de cycle de vie Deploy de la couche PHP App Server. Lorsque vous déployez SimplePHPApp, AWS OpsWorks Stacks s'exécute`appsetup.rb`, ce qui crée un fichier de configuration nommé `db-connect.php` qui définit la connexion, comme indiqué dans l'extrait suivant.

```
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
```

Les variables qui caractérisent la connexion (, `hostuser`, etc.) reçoivent les valeurs correspondantes à partir des attributs du JSON de `[:deploy][:app_name][:database]` déploiement. Pour des raisons de simplicité, l'exemple suppose que vous ayez déjà créé une table nommée `urler` et que, par conséquent, le nom de la table soit représenté par `[:phpapp][:dbtable]` dans le fichier d'attributs du livre de recettes.

Cette recette permet en fait de connecter le serveur d'applications PHP à n'importe quel serveur de base de données MySQL, et pas seulement aux membres d'une couche MySQL. Pour utiliser un autre serveur MySQL, il vous suffit de définir les `[:database]` attributs sur des valeurs adaptées à votre serveur, ce que vous pouvez faire en utilisant du JSON personnalisé. AWS OpsWorks Stacks intègre ensuite ces attributs et valeurs dans la configuration de la pile et les attributs de déploiement et les `appsetup.rb` utilise pour créer le modèle qui établit la connexion. Pour plus d'informations sur le remplacement du JSON de configuration et de déploiement de la pile, consultez Remplacement des attributs.

Comment connecter une instance de serveur d'applications à Amazon RDS

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients

de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section décrit comment personnaliser Mise en route des piles Linux Chef 11 pour que le serveur MyStack d'applications PHP se connecte à une instance RDS.

Rubriques

- Création d'une base de données Amazon RDS MySQL
- Personnaliser la pile pour se connecter à la base de données RDS

Création d'une base de données Amazon RDS MySQL

Important

Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous êtes maintenant prêt à créer une base de données RDS pour cet exemple à l'aide de l'assistant de lancement d'instance de base de données de la console Amazon RDS. La procédure suivante est un bref résumé des détails essentiels. Pour une description détaillée de la création d'une base de données, consultez Mise en route avec Amazon RDS.

Pour créer la base de données Amazon RDS

1. Si vous créez une base de données RDS pour la première fois, cliquez sur Get Started Now. Sinon, cliquez sur RDS Dashboard dans le volet de navigation, puis cliquez sur Launch a DB Instance.
2. Sélectionnez MySQL Community Edition comme instance DB.
3. Sous la question Envisagez-vous d'utiliser cette base de données à des fins de production ?, sélectionnez Non, cette instance..., ce qui suffit pour l'exemple. En production, vous pouvez

sélectionner Yes, use Multi-AZ Deployment... (Oui, utilisez le déploiement multi-AZ). Cliquez sur Étape suivante.

4. Sur la page Spécification de détails de base de données, indiquez les valeurs suivantes :

   - Classe d'instance DB : db.t2.micro
   - Déploiement multi-AZ : Non
   - Stockage alloué : **5** Go
   - Identifiant de l'instance DB : **rdsexample**
   - Identifiant principal : **opsworksuser**
   - Master Password : spécifiez un mot de passe approprié et enregistrez-le en vue d'une utilisation ultérieure.

   Acceptez les valeurs par défaut pour les autres options et cliquez sur Next Step.

5. Sur la page Configuration de paramètres avancés, spécifiez les valeurs suivantes :

   - Dans la section Network & Security, pour VPC Security Group(s), sélectionnez phpsecgroup (VPC).
   - Dans la section Database options (Options de la base de données), pour Database Name (Nom de la base de données), saisissez **rdsexampledb**.
   - Dans la section Backup, définissez Backup Retention Period sur 0 dans le cas de cette procédure pas à pas.

   Acceptez les paramètres par défaut pour les autres options et cliquez sur Launch DB Instance.

6. Choisissez View Your DB Instances pour voir la liste des instances DB.
7. Sélectionnez l'instance rdsexample dans la liste et cliquez sur la flèche pour révéler le point de terminaison de l'instance et autres détails. Enregistrez le point de terminaison en vue d'une utilisation ultérieure. Il se présente sous la forme `rdsexample.c6c8mntzhgv0.us-west-2.rds.amazonaws.com:3306`. Enregistrez juste le nom DNS ; vous n'aurez pas besoin du numéro de port.
8. Utilisez un outil tel que MySQL Workbench pour créer une table nommée `urler` dans la base de données `rdsexampledb` en utilisant la commande SQL suivante :

```
CREATE TABLE urler(id INT UNSIGNED NOT NULL AUTO_INCREMENT,author VARCHAR(63) NOT
 NULL,message TEXT,PRIMARY KEY (id))
```

Personnaliser la pile pour se connecter à la base de données RDS

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Une fois que vous avez créé une instance RDS à utiliser comme base de données principale pour le serveur d'applications PHP, vous pouvez la personnaliser MyStack à partir de. Mise en route des piles Linux Chef 11

Pour connecter le serveur d'applications PHP à une base de données RDS

1. Ouvrez la console AWS OpsWorks Stacks et créez une pile avec une couche PHP App Server contenant une instance et déployez SimplePHPApp, comme décrit dansMise en route des piles Linux Chef 11. Cette pile utilise la version 1 de SimplePHPApp, qui n'utilise pas de connexion à la base de données.
2. Mettez à jour la configuration de la pile pour utiliser les livres personnalisés qui incluent la recette `appsetup.rb`, et les fichiers modèles et les fichiers d'attributs associés.

    1. Définissez Utiliser les livres de recettes Chef personnalisés sur Oui.
    2. Définissez Repository type sur Git et Repository URL sur `git://github.com/amazonwebservices/opsworks-example-cookbooks.git`.
3. Ajoutez les éléments suivants à la zone Custom Chef JSON de la pile pour affecter les données de connexion RDS aux attributs `[:database]` qu'`appsetup.rb` utilise pour créer le fichier de configuration.

```
{
  "deploy": {
    "simplephpapp": {
      "database": {
        "username": "opsworksuser",
        "password": "your_password",
        "database": "rdsexampledb",
```

```
        "host": "rds_endpoint",
        "adapter": "mysql"
      }
    }
  }
}
```

Utilisez les valeurs d'attributs suivantes :

- username : nom d'utilisateur principal que vous avez spécifié lorsque vous avez créé l'instance RDS.

  Cet exemple utilise `opsworksuser`.
- password : mot de passe que vous avez spécifié lorsque vous avez créé l'instance RDS.

  Remplissez le mot de passe que vous avez spécifié.
- database : base de données que vous avez créée lorsque vous avez créé l'instance RDS.

  Cet exemple utilise `rdsexampledb`.
- host : point de terminaison de l'instance RDS, que vous avez obtenu à partir de la console RDS lorsque vous avez créé l'instance dans la section précédente. N'incluez pas le numéro de port.
- adapter : carte.

  Comme l'instance RDS de cet exemple utilise MySQL, adapter a la valeur `mysql`. Contrairement aux autres attributs, adapter n'est pas utilisé par `appsetup.rb`. Il est plutôt utilisé par la recette de configuration intégrée de la couche PHP App Server pour créer un fichier de configuration différent.

4. Modifiez la PHPApp configuration Simple pour spécifier une version de Simple PHPApp qui utilise une base de données principale, comme suit :

   - Document root : définissez l'option sur `web`.
   - Branch/Revision : définissez l'option sur `version2`.

   Laissez les autres options inchangées.
5. Modifiez la couche PHP App Server pour configurer la connexion à la base de données en l'ajoutant `phpapp::appsetup` aux recettes de déploiement de la couche.

6. Déployez la nouvelle PHPApp version Simple.
7. Lorsque Simple PHPApp est déployé, exécutez l'application en accédant à la page Instances et en cliquant sur l'adresse IP publique de l'instance php-app1. Vous devez voir la page suivante dans votre navigateur, ce qui vous permet de saisir du texte et de le stocker dans la base de données.



> **Note**
>
> Si votre pile possède une couche MySQL, AWS OpsWorks Stacks attribue automatiquement les données de connexion correspondantes aux `[:database]` attributs. Cependant, si vous affectez le JSON personnalisé à la pile qui définit différentes valeurs `[:database]`, celles-ci remplacent les valeurs par défaut. Comme les `[:deploy]` attributs sont installés sur chaque instance, toutes les recettes qui dépendent des `[:database]` attributs utiliseront les données de connexion personnalisées, et non les données de la couche MySQL pour le. Si vous souhaitez qu'une couche de serveur d'applications particulière utilise les données de connexion personnalisées, attribuez le JSON personnalisé à l'événement Deploy de la couche et limitez ce déploiement à cette couche. Pour plus d'informations sur l'utilisation des attributs de déploiement, consultez Déploiement d'applications. Pour plus d'informations sur la substitution des attributs AWS OpsWorks Stacks intégrés, consultez Remplacement des attributs.

Utiliser ElastiCache Redis comme magasin de valeurs clés en mémoire

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Note

Cette rubrique est basée sur une pile Linux, mais les piles Windows peuvent également utiliser Amazon ElastiCache (ElastiCache). Pour un exemple d'utilisation ElastiCache avec une instance Windows, consultez ElastiCache Asp.NET Session Store.

Vous pouvez souvent améliorer les performances du serveur d'applications en utilisant un serveur de mise en cache pour fournir un stockage clé-valeur en mémoire pour de petits éléments de données tels que des chaînes. Amazon ElastiCache est un service AWS qui facilite la prise en charge de la mise en cache pour votre serveur d'applications, à l'aide des moteurs de mise en cache Memcached ou Redis. AWS OpsWorks Stacks fournit un support intégré pour Memcached. Toutefois, si Redis répond mieux à vos besoins, vous pouvez personnaliser votre stack afin que vos serveurs d'applications utilisent ElastiCache Redis.

Cette rubrique décrit le processus de base de la prise en charge de la mise en cache ElastiCache Redis pour les piles Linux, en utilisant un serveur d'applications Rails comme exemple. Il part du principe que vous avez déjà une application Ruby on Rails appropriée. Pour plus d'informations ElastiCache, consultez Qu'est-ce qu'Amazon ElastiCache ? .

Rubriques

- Étape 1 : créer un cluster ElastiCache Redis
- Étape 2 : Configurer une pile Rails
- Étape 3 : Créer et déployer un livre de recettes personnalisé
- Étape 4 : Attribuer la recette à un LifeCycle événement

- Étape 5 : Ajouter les informations d'accès au JSON de configuration de la pile
- Étape 6 : Déployer et exécuter l'application

Étape 1 : créer un cluster ElastiCache Redis

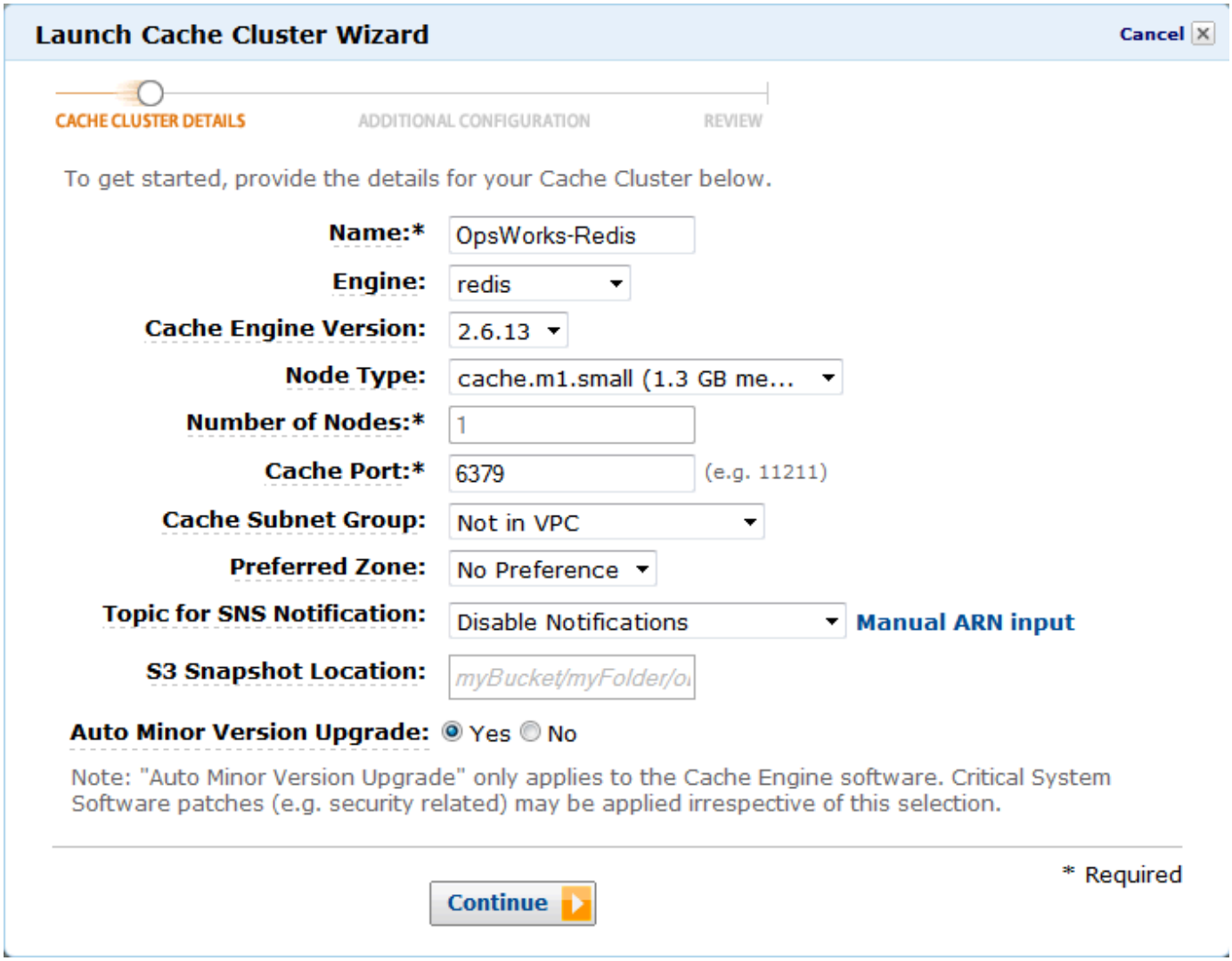
> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous devez d'abord créer un cluster Amazon ElastiCache Redis à l'aide de la ElastiCache console, de l'API ou de la CLI. Ce qui suit décrit comment utiliser la console pour créer un cluster.

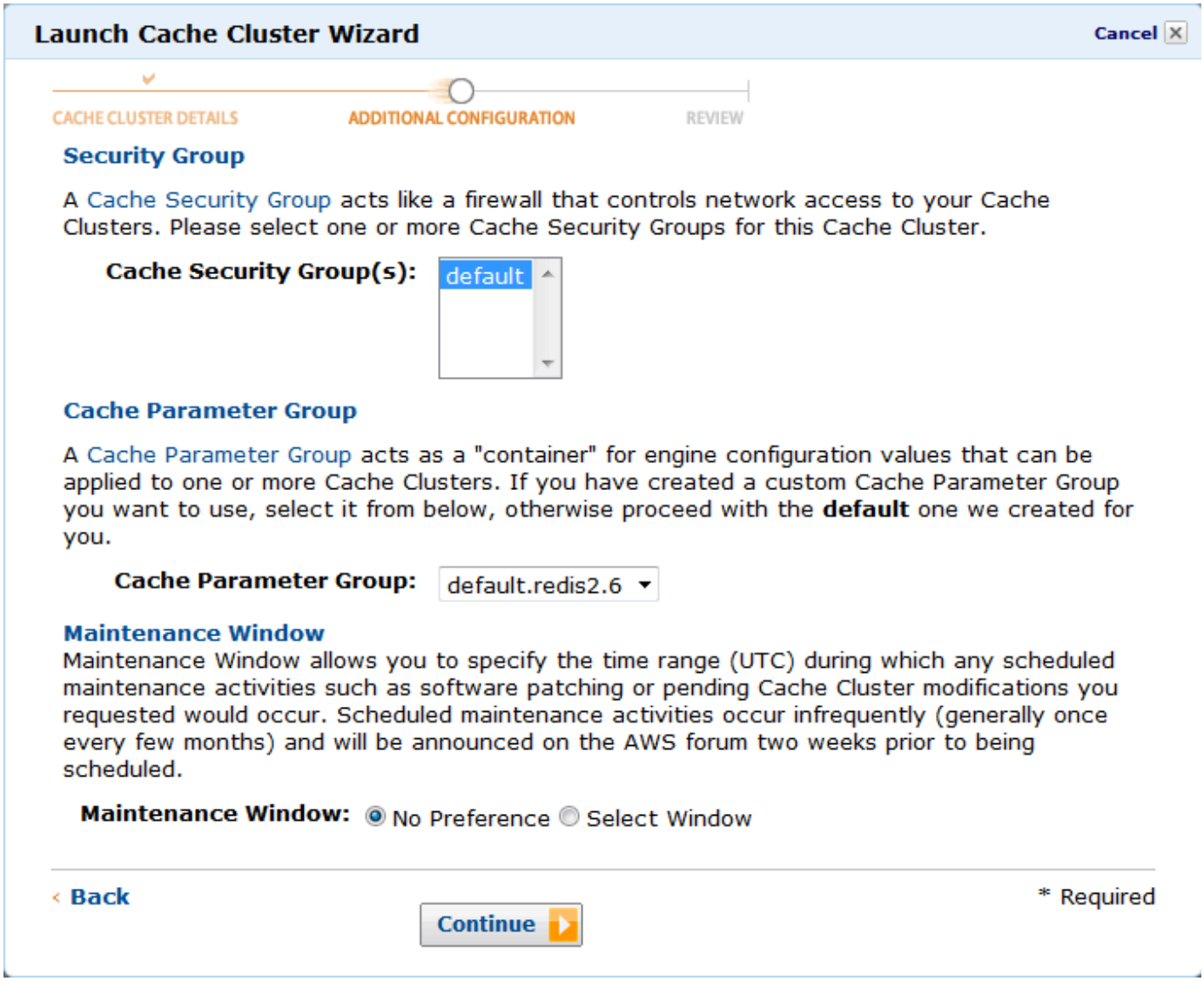
Pour créer un cluster ElastiCache Redis

1. Accédez à la console ElastiCache et cliquez sur Lancer le cluster de cache pour démarrer l'assistant de lancement de cluster de cache.
2. Sur la page Cache Cluster Details, procédez comme suit :
   - Définissez Nom avec le nom de votre serveur de cache.

     Cet exemple utilise OpsWorks -Redis.
   - Définissez Engine (Moteur) sur redis.
   - Définissez Rubrique pour notification SNS sur Désactiver les notifications.
   - Acceptez les valeurs par défaut pour les autres paramètres, puis cliquez sur Continuer.



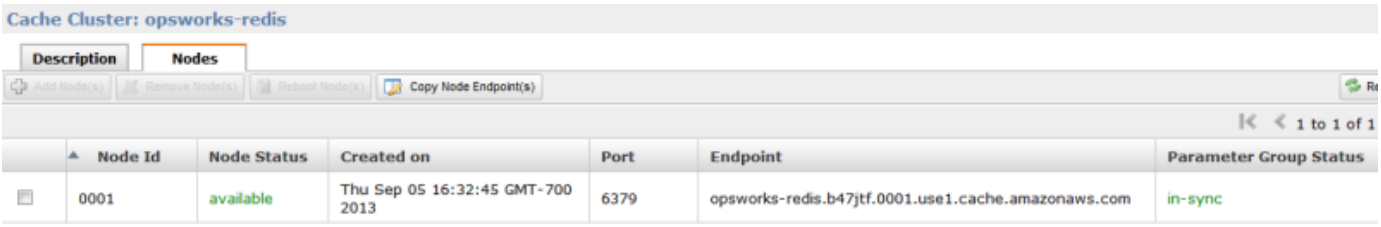
3. Sur la page Configuration supplémentaire, acceptez les valeurs par défaut et cliquez sur Continuer.



4. Cliquez sur Lancer le cluster de cache pour créer le cluster.

> **Important**
>
> Le groupe de sécurité du cache par défaut est suffisant pour cet exemple, mais en production, vous devez en créer un qui convient à votre environnement. Pour plus d'informations, consultez Gestion des groupes de sécurité du cache.

5. Une fois que le cluster a démarré, cliquez sur le nom pour ouvrir la page des détails, puis cliquez sur l'onglet Nodes. Enregistrez les valeurs Port et Endpoint du cluster pour une utilisation ultérieure.



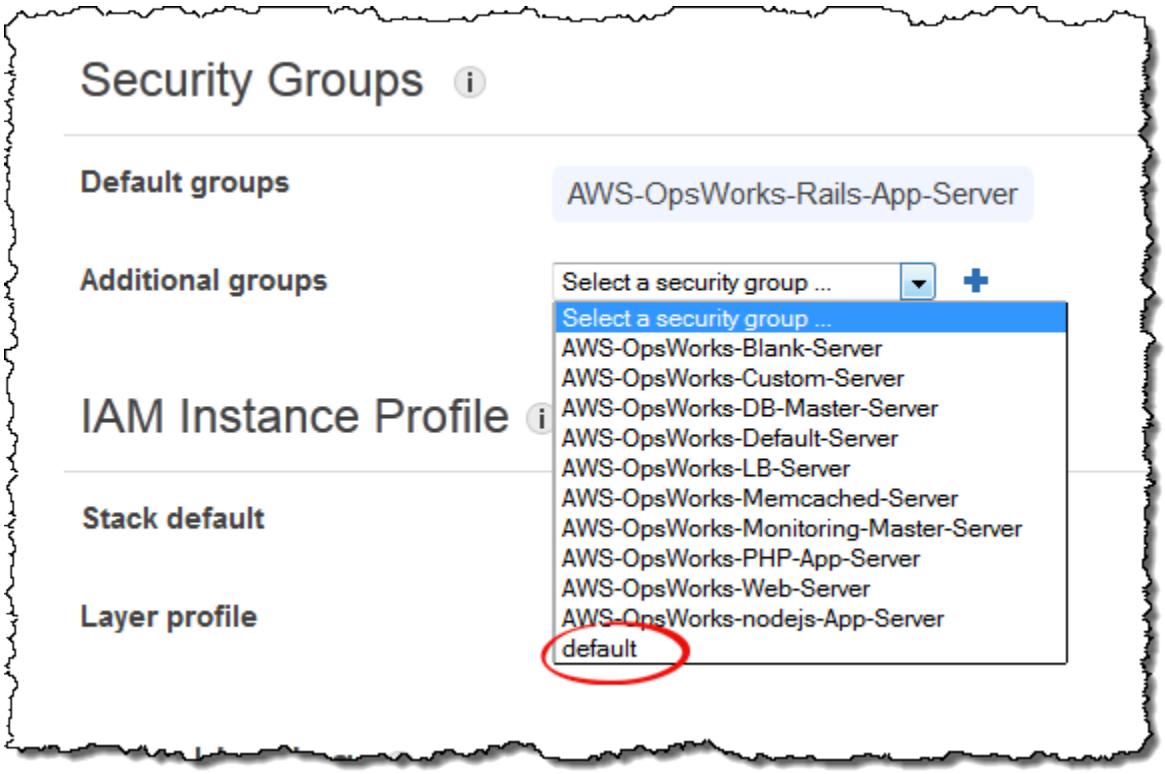
Étape 2 : Configurer une pile Rails

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Outre la création d'une pile prenant en charge une couche Rails App Server, vous devez également configurer les groupes de sécurité de la couche afin que le serveur Rails puisse communiquer correctement avec le serveur Redis.

Pour configurer une pile

1. Créez une nouvelle pile, nommée d'après cet **`RedisStack`** exemple, et ajoutez une couche Rails App Server. Vous pouvez utiliser les valeurs par défaut pour les deux. Pour plus d'informations, consultez Créer une pile et Création d'une OpsWorks couche .
2. Sur la page Layers, pour Rails App Server, cliquez sur Sécurité, puis sur Modifier.
3. Accédez à la section Groupe de sécurité et ajoutez le groupe de sécurité du cluster ElastiCache à Groupes supplémentaires. Dans cet exemple, sélectionnez le groupe de sécurité par défaut, cliquez sur + pour l'ajouter à la couche et cliquez sur Enregistrer pour enregistrer la nouvelle configuration.



4. Ajoutez une instance à la couche Rails App Server et démarrez-la. Pour plus d'informations sur la façon d'ajouter et de démarrer des instances, consultez Ajout d'une instance à une couche.

Étape 3 : Créer et déployer un livre de recettes personnalisé

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

En l'état, la pile n'est pas encore assez fonctionnelle ; vous devez activer votre application pour accéder au serveur Redis. L'approche la plus souple consiste à placer un fichier YAML avec les informations d'accès dans le sous-dossier `config` de l'application. L'application peut ensuite obtenir les informations du fichier. A l'aide de cette approche, vous pouvez modifier les informations de connexion sans réécrire et redéployer l'application. Pour cet exemple, le fichier doit être nommé `redis.yml` et contenir le nom d'hôte et le port du cluster ElastiCache , comme suit :

```
host: cache-cluster-hostname
port: cache-cluster-port
```

Vous pouvez copier manuellement ce fichier sur vos serveurs, mais une meilleure approche consiste à implémenter une recette Chef pour générer le fichier, et à demander à AWS OpsWorks Stacks d'exécuter la recette sur chaque serveur. Les recettes Chef sont des applications Ruby spécialisées que AWS OpsWorks Stacks utilise pour effectuer des tâches sur des instances telles que l'installation de packages ou la création de fichiers de configuration. Les recettes sont rassemblées dans un livre de recettes, qui peut contenir plusieurs recettes et fichiers apparentés, tels que les modèles des fichiers de configuration. Le livre de recettes est placé dans un référentiel, tel que GitHub, et doit avoir une structure de répertoire standard. Si vous n'avez pas encore un référentiel de livres de recettes personnalisé, consultez Référentiels de livres de recettes pour plus d'informations sur la façon d'en configurer un.

Pour cet exemple, ajoutez un livre de recettes nommé `redis-config` à votre référentiel de livres de recettes avec le contenu suivant :

```
my_cookbook_repository
  redis-config
    recipes
      generate.rb
    templates
      default
        redis.yml.erb
```

Le dossier `recipes` contient une recette `generate.rb`, qui génère le fichier de configuration de l'application à partir de `redis.yml.erb`, comme suit :

```
node[:deploy].each do |app_name, deploy_config|
  # determine root folder of new app deployment
  app_root = "#{deploy_config[:deploy_to]}/current"

  # use template 'redis.yml.erb' to generate 'config/redis.yml'
  template "#{app_root}/config/redis.yml" do
    source "redis.yml.erb"
    cookbook "redis-config"
```

```
    # set mode, group and owner of generated file
    mode "0660"
    group deploy_config[:group]
    owner deploy_config[:user]

    # define variable "@redis" to be used in the ERB template
    variables(
      :redis => deploy_config[:redis] || {}
    )

    # only generate a file if there is Redis configuration
    not_if do
      deploy_config[:redis].blank?
    end
  end
end
```

La recette dépend des données issues de la configuration de la pile AWS OpsWorks Stacks et de l'objet JSON de déploiement, qui est installé sur chaque instance et contient des informations détaillées sur la pile et les applications déployées. Le nœud `deploy` de l'objet a la structure suivante :

```
{
   ...
  "deploy": {
    "app1": {
      "application" : "short_name",
      ...
    }
    "app2": {
      ...
    }
    ...
  }
}
```

Le nœud Deploy contient un ensemble d'objets JSON intégrés, un pour chaque application déployée et nommé d'après le nom court de l'application. Chaque objet d'application contient un ensemble d'attributs qui définissent la configuration de l'application, tels que la racine du document et le type d'application. Pour obtenir la liste des attributs deploy, consultez Attributs deploy. Les recettes

peuvent utiliser la syntaxe d'attribut Chef pour représenter les valeurs JSON de déploiement et de configuration de la pile. Par exemple, `[:deploy][:app1][:application]` représente le nom court de l'application app1.

Pour chaque application dans `[:deploy]`, la recette exécute le bloc de code associé, où `deploy_config` représente l'attribut de l'application. La recette définit d'abord `app_root` sur le répertoire racine de l'application, `[:deploy][:app_name][:deploy_to]/current`. Elle utilise ensuite une ressource de modèle Chef pour générer un fichier de configuration à partir de `redis.yml.erb` et le placer dans le fichier `app_root/config`.

Les fichiers de configuration sont généralement créés à partir de modèles, avec la plupart si ce n'est la totalité des paramètres définis par les attributs Chef. Avec les attributs, vous pouvez modifier les paramètres à l'aide d'un JSON personnalisé, comme décrit ultérieurement, au lieu de réécrire le fichier modèle. Le modèle `redis.yml.erb` contient les éléments suivants :

```
host: <%= @redis[:host] %>
port: <%= @redis[:port] || 6379 %>
```

Les éléments <%... %> sont des espaces réservés qui représentent une valeur d'attribut.

- `<%= @redis[:host] %>` représente la valeur de `redis[:host]`, qui correspond au nom d'hôte du cluster de cache.
- `<%= @redis[:port] || 6379 %>` représente la valeur de l'attribut `redis[:port]` ou, si cet attribut n'est pas défini, la valeur du port par défaut, 6379.

La ressource `template` fonctionne comme suit :

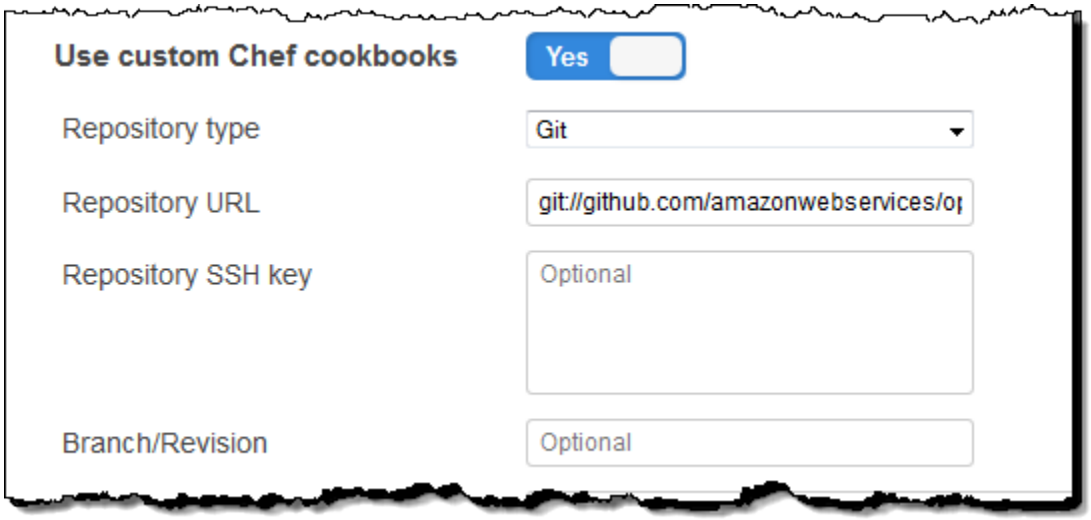
- `source` et `cookbook` spécifient respectivement les noms du modèle et du livre de recettes.
- `mode`, `group` et `owner` donnent au fichier de configuration les mêmes droits d'accès que l'application.
- La section `variables` définit la variable `@redis` utilisée dans le modèle avec la valeur de l'attribut `[:redis]` de l'application.

  Les valeurs de l'attribut `[:redis]` sont définies en utilisant un JSON personnalisé, comme décrit plus tard ; ce n'est pas l'un des attributs standard de l'application.
- La directive `not_if` garantit que la recette ne génère pas un fichier de configuration s'il en existe déjà un.

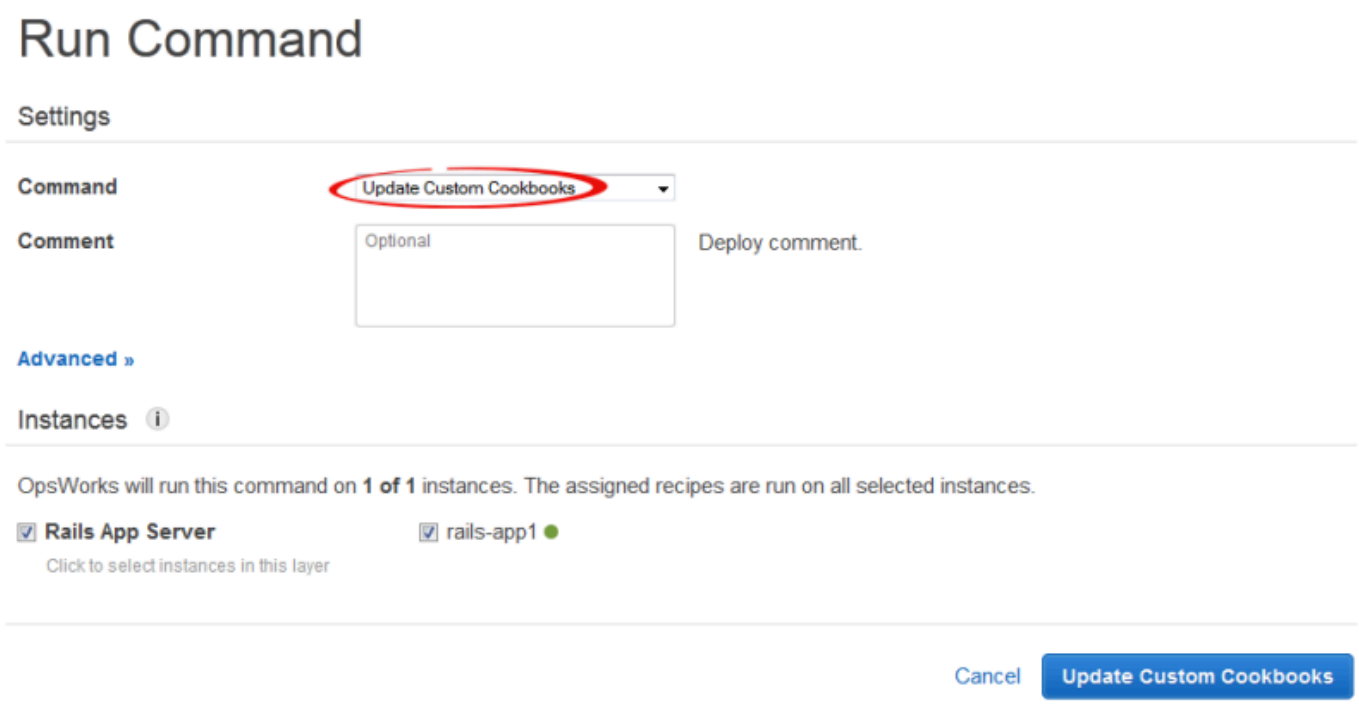
Une fois que vous avez créé le livre de recettes, vous devez le déployer sur le cache du livre de recettes de chaque instance. Cette opération n'exécute pas la recette ; elle installe simplement le nouveau livre de recettes sur les instances de la pile. Généralement, vous exécutez une recette en l'affectant à l'un des événements de cycle de vie d'une couche, comme décrit plus tard.

Pour déployer votre livre de recettes personnalisé

1. Sur la page AWS OpsWorks Stacks Stack, cliquez sur Stack Settings, puis sur Modifier.
2. Dans la section Gestion de la configuration, définissez Utiliser les livres de recettes Chef personnalisés sur Oui, saisissez les informations du référentiel du livre de recettes et cliquez sur Enregistrer pour mettre à jour la configuration de la pile.



3. Sur la page Pile, cliquez sur Run Command (Exécuter la commande), sélectionnez la commande de pile Mettre à jour les livres de recettes personnalisées, puis cliquez sur Mettre à jour les livres de recettes personnalisées pour installer le nouveau livre de recettes dans les caches de livre de recettes de l'instance.



Si vous modifiez votre livre de recettes, il suffit d'exécuter à nouveau Mettre à jour les livres de recettes personnalisées pour installer la version mise à jour. Pour plus d'informations sur cette procédure, consultez Installation de livres de recettes personnalisés.

Étape 4 : Attribuer la recette à un LifeCycle événement

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

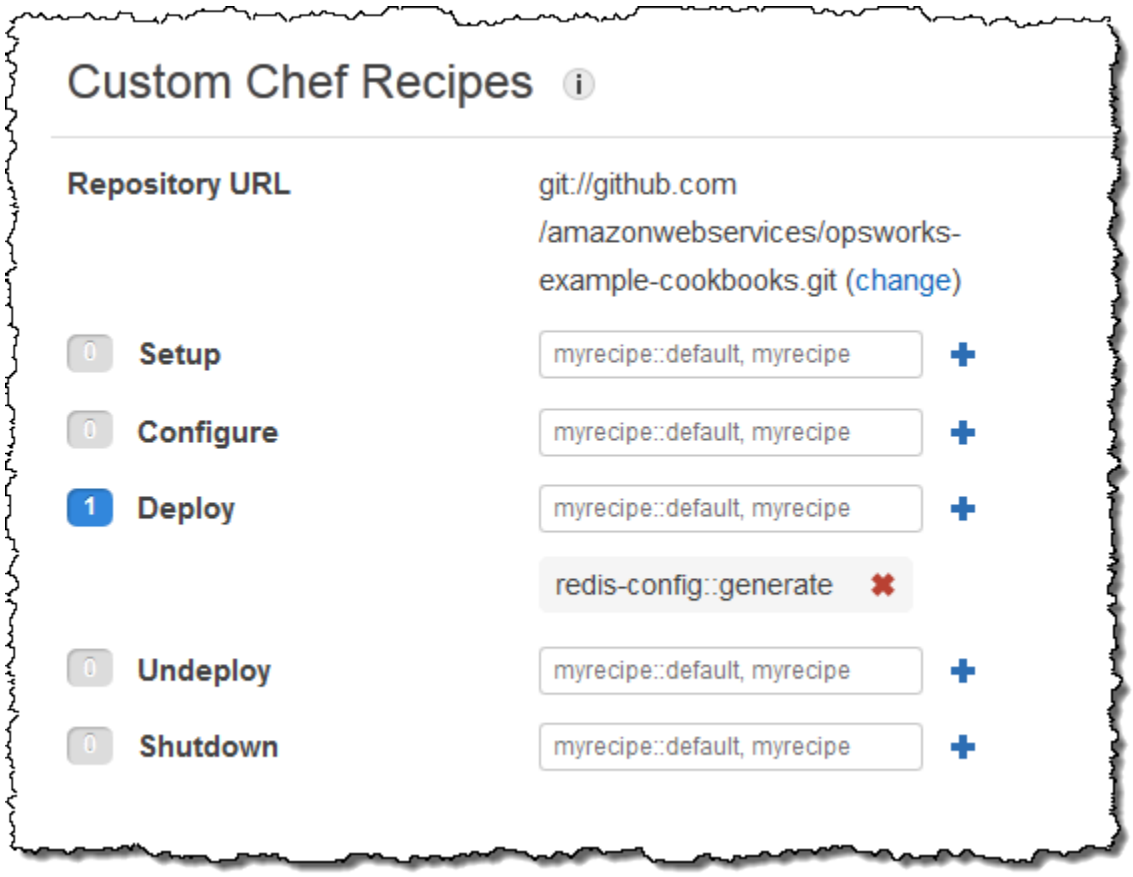
Vous pouvez exécuter des recettes personnalisées manuellement, mais la meilleure approche consiste généralement à faire en sorte que AWS OpsWorks Stacks les exécute automatiquement. Chaque couche possède un ensemble de recettes intégrées associées à chacun des cinq événements du cycle de vie : installation, configuration, déploiement, annulation du déploiement et arrêt. Chaque fois qu'un événement se produit pour une instance, AWS OpsWorks Stacks exécute les recettes associées pour chacune des couches de l'instance, lesquelles gèrent les tâches

correspondantes. Par exemple, lorsqu'une instance finit de démarrer, AWS OpsWorks Stacks déclenche un événement de configuration. Cet événement exécute les recettes Setup de la couche associée, qui généralement gèrent les tâches telles que l'installation et la configuration de packages.

Vous pouvez demander à AWS OpsWorks Stacks d'exécuter une recette personnalisée sur les instances d'une couche en affectant la recette à l'événement du cycle de vie approprié. Pour cet exemple, vous devez attribuer la `generate.rb` recette à l'événement Deploy de la couche Rails App Server. AWS OpsWorks Stacks l'exécutera ensuite sur les instances de la couche au démarrage, une fois les recettes de configuration terminées et à chaque fois que vous déployez une application. Pour de plus amples informations, veuillez consulter Exécution automatique des recettes.

Pour attribuer une recette à l'événement Deploy de la couche Rails App Server

1. Sur la page AWS OpsWorks Stacks Layers, pour Rails App Server, cliquez sur Recettes, puis sur Modifier.
2. Sous Recettes Chef personnalisées, ajoutez le nom complet de la recette à l'événement Deploy (Déployer) et cliquez sur +. Un nom de recette complet utilise le format *`cookbookname`*`::`*`recipename`* , où *`recipename`* n'inclut pas l'extension `.rb`. Pour cet exemple, le nom complet est `redis-config::generate`. Puis, cliquez sur Enregistrer pour mettre à jour la configuration de la couche.



Étape 5 : Ajouter les informations d'accès au JSON de configuration de la pile

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

La recette `generate.rb` repose sur une paire d'attributs JSON de configuration et de déploiement de la pile qui représentent le nom d'hôte et le port du serveur Redis. Bien que ces attributs fassent partie de l'espace de `[:deploy]` noms standard, ils ne sont pas automatiquement définis par AWS OpsWorks Stacks. Au lieu de cela, vous définissez les attributs et leurs valeurs en ajoutant un objet JSON personnalisé à la pile. L'exemple suivant illustre le JSON personnalisé pour cet exemple.

Pour ajouter les informations d'accès au JSON de configuration et de déploiement de la pile

1. Sur la page AWS OpsWorks Stacks Stack, cliquez sur Stack Settings, puis sur Modifier.
2. Dans la section Configuration Management, ajoutez les informations d'accès à la zone Custom Chef JSON. Elle doit ressembler à l'exemple suivant, avec ces modifications :

   - Remplacez `elasticache_redis_example` par le nom court de votre application.
   - Remplacez les `port` valeurs `host` et par les valeurs de l'instance de serveur ElastiCache Redis que vous avez créée dansÉtape 1 : créer un cluster ElastiCache Redis.

```
{
  "deploy": {
     "elasticache_redis_example": {
       "redis": {
         "host": "mycluster.XXXXXXXXX.amazonaws.com",
         "port": "6379"
       }
     }
  }
}
```



L'avantage de cette approche est que vous pouvez modifier la valeur du port ou de l'hôte à tout moment sans toucher à votre livre de recettes personnalisé. AWS OpsWorks Stacks fusionne le JSON personnalisé dans le JSON intégré et l'installe sur les instances de la pile pour tous les

événements du cycle de vie ultérieurs. Les applications peuvent ensuite accéder aux valeurs d'attribut en utilisant la syntaxe de nœud Chef, comme décrit dans Étape 3 : Créer et déployer un livre de recettes personnalisé. La prochaine fois où vous déployez une application, AWS OpsWorks Stacks installe un JSON de configuration et de déploiement de la pile, qui contient les nouvelles définitions et `generate.rb` crée un fichier de configuration avec les valeurs de port et d'hôte mises à jour.

> **Note**
>
> Comme `[:deploy]` inclut automatiquement un attribut pour chaque application déployée, `[:deploy][elasticache_redis_example]` est déjà dans le JSON de configuration et de déploiement de la pile. Cependant, il `[:deploy][elasticache_redis_example]` n'inclut pas d'`[:redis]`attribut. Le fait de les définir avec du JSON personnalisé indique à AWS OpsWorks Stacks d'y ajouter ces attributs. `[:deploy][elasticache_redis_example]` Vous pouvez aussi utiliser un JSON personnalisé pour remplacer les attributs existants. Pour de plus amples informations, veuillez consulter Remplacement des attributs.

Étape 6 : Déployer et exécuter l'application

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cet exemple suppose que vous avez une application Ruby on Rails qui utilise Redis. Pour accéder au fichier de configuration, vous pouvez ajouter le GEM `redis` à votre fichier Gemfile et créer un initialiseur Rails dans `config/initializers/redis.rb`, comme suit :

```
REDIS_CONFIG = YAML::load_file(Rails.root.join('config', 'redis.yml'))
$redis = Redis.new(:host => REDIS_CONFIG['host'], :port => REDIS_CONFIG['port'])
```

Créez ensuite une application pour représenter votre application et déployez-la sur les instances de la couche Rails App Server, qui met à jour le code de l'application et s'exécute `generate.rb` pour générer le fichier de configuration. Lorsque vous exécutez l'application, elle utilise l'instance ElastiCache Redis comme magasin de valeurs-clés en mémoire.

Utilisation d'un compartiment Amazon S3

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les applications utilisent souvent un compartiment Amazon Simple Storage Service (Amazon S3) pour stocker des éléments volumineux tels que des images ou d'autres fichiers multimédia. Bien que AWS OpsWorks Stacks ne fournisse pas de support intégré pour Amazon S3, vous pouvez facilement personnaliser une pile pour permettre à votre application d'utiliser le stockage Amazon S3. Cette rubrique décrit le processus de base qui consiste à fournir un accès aux applications à Amazon S3, en utilisant une pile Linux avec un serveur d'applications PHP par exemple. Les principes de base s'appliquent également aux piles Windows.

Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Rubriques

- Étape 1 : Créer un compartiment Amazon S3
- Étape 2 : Créer une pile de serveurs d'applications PHP
- Étape 3 : Créer et déployer un livre de recettes personnalisé
- Étape 4 : Attribuer les recettes aux LifeCycle événements
- Étape 5 : Ajouter les informations d'accès aux attributs de configuration et de déploiement de la pile
- Étape 6 : Déploiement et exécution PhotoApp

Étape 1 : Créer un compartiment Amazon S3

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous devez d'abord créer un compartiment Amazon S3. Vous pouvez le faire directement à l'aide de la console, de l'API ou de la CLI Amazon S3, mais un moyen plus simple de créer des ressources consiste souvent à utiliser un AWS CloudFormation modèle. Le modèle suivant crée un compartiment Amazon S3 pour cet exemple et définit un profil d'instance avec un rôle IAM qui accorde un accès illimité au compartiment. Vous pouvez ensuite utiliser un paramètre de couche pour attacher le profil d'instance aux instances serveur d'applications de la pile, ce qui permet à l'application d'accéder au compartiment, comme décrit ultérieurement. L'utilité des profils d'instance ne se limite pas à Amazon S3 ; ils sont utiles pour intégrer divers services AWS.

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "AppServerRootRole": {
         "Type": "AWS::IAM::Role",
         "Properties": {
            "AssumeRolePolicyDocument": {
               "Statement": [ {
                  "Effect": "Allow",
                  "Principal": {
                     "Service": [ "ec2.amazonaws.com" ]
                  },
                  "Action": [ "sts:AssumeRole" ]
               } ]
            },
            "Path": "/"
         }
      },
      "AppServerRolePolicies": {
```

```
         "Type": "AWS::IAM::Policy",
         "Properties": {
            "PolicyName": "AppServerS3Perms",
            "PolicyDocument": {
               "Statement": [ {
                  "Effect": "Allow",
                  "Action": "s3:*",
                  "Resource": { "Fn::Join" : ["", [ "arn:aws:s3:::", { "Ref" :
 "AppBucket" } , "/*" ]
                  ] }
               } ]
            },
            "Roles": [ { "Ref": "AppServerRootRole" } ]
         }
      },
      "AppServerInstanceProfile": {
         "Type": "AWS::IAM::InstanceProfile",
         "Properties": {
            "Path": "/",
            "Roles": [ { "Ref": "AppServerRootRole" } ]
         }
      },
     "AppBucket" : {
      "Type" : "AWS::S3::Bucket"
      }
   },
   "Outputs" : {
       "BucketName" : {
           "Value" : { "Ref" : "AppBucket" }
       },
       "InstanceProfileName" : {
           "Value" : { "Ref" : "AppServerInstanceProfile" }
       }
   }
}
```

Plusieurs choses se produisent lorsque vous lancez le modèle :

- La `AWS::S3::Bucket` ressource crée un compartiment Amazon S3.
- La ressource `AWS::IAM::InstanceProfile` crée un profil d'instance qui sera attribué aux instances serveur d'applications.
- La ressource `AWS::IAM::Role` crée le rôle du profil d'instance.

- La `AWS::IAM::Policy` ressource définit les autorisations du rôle afin de permettre un accès illimité aux compartiments Amazon S3.
- La section `Outputs` affiche les noms des profils d'instance et des compartiments dans la console AWS CloudFormation une fois que vous avez lancé le modèle.

  Vous aurez besoin de ces valeurs pour configurer votre pile et vos applications.

Pour plus d'informations sur la création de AWS CloudFormation modèles, voir Apprendre les bases des modèles.

Pour créer un compartiment Amazon S3

1. Copiez l'exemple de modèle sur un fichier texte de votre système.

   Cet exemple suppose que le fichier se nomme `appserver.template`.
2. Ouvrez la console AWS CloudFormation et choisissez Créer une pile.
3. Dans la zone Nom de la pile, saisissez le nom de la pile.

   Cet exemple suppose que le fichier se nomme **`AppServer`**.
4. Choisissez Upload template file (Charger un fichier de modèle), choisissez Parcourir, sélectionnez le fichier `appserver.template` que vous avez créé à l'étape 1, puis choisissez Étape suivante.
5. Sur la page Specify Parameters (Spécifier des paramètres), sélectionnez I acknowledge that this template may create IAM resources (Je confirme que ce modèle peut créer des ressources IAM), puis choisissez Étape suivante sur chaque page de l'Assistant jusqu'à ce que vous atteigniez la fin. Choisissez Créer.
6. Une fois que la AppServerpile a atteint le statut CREATE_COMPLETE, sélectionnez-la et choisissez l'onglet Sorties.

   Il se peut que vous ayez besoin d'actualiser plusieurs fois pour mettre à jour le statut.
7. Dans l'onglet Sorties, enregistrez les InstanceProfileNamevaleurs BucketNameet pour une utilisation ultérieure.

> **Note**
>
> AWS CloudFormation utilise le terme pile pour désigner l'ensemble de ressources créées à partir d'un modèle ; ce n'est pas la même chose qu'une pile AWS OpsWorks Stacks.

Étape 2 : Créer une pile de serveurs d'applications PHP

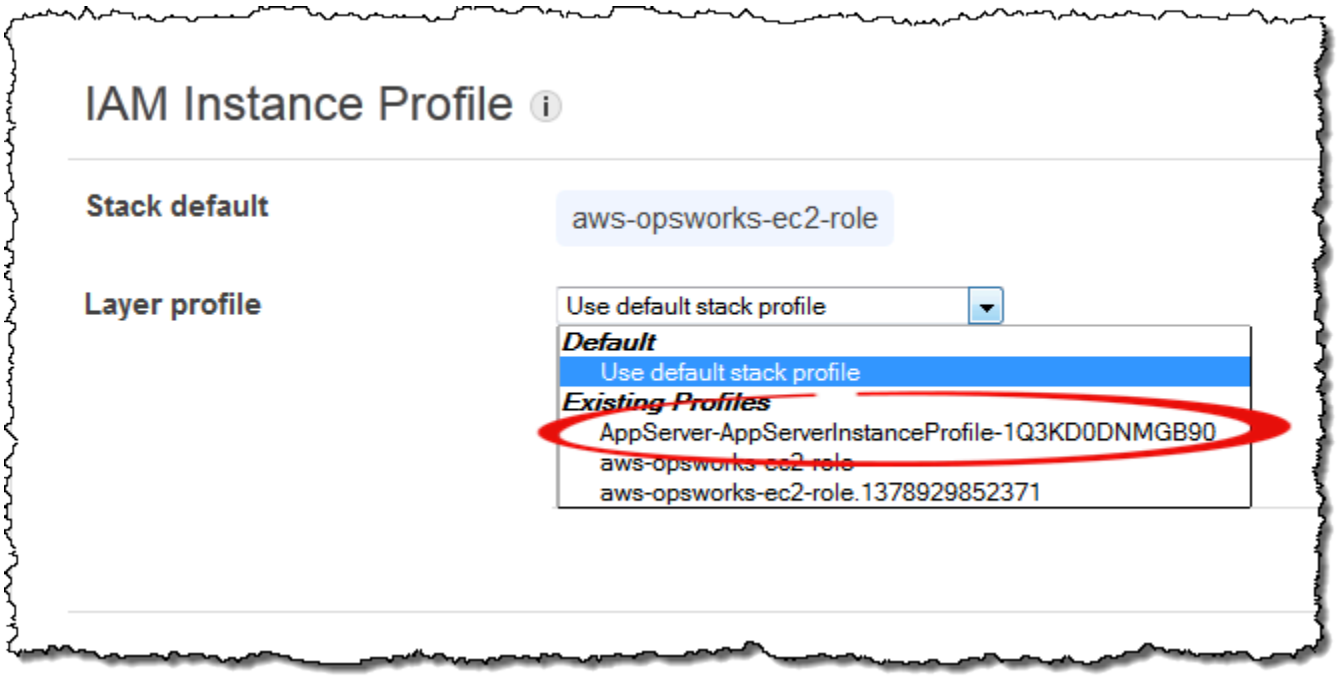
> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

La pile se compose de deux couches, PHP App Server et MySQL, chacune avec une instance. L'application stocke les photos dans un compartiment Amazon S3, mais utilise l'instance MySQL comme magasin de données principal pour conserver les métadonnées de chaque photo.

Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Pour créer la pile

1. Créez une nouvelle pile, nommée d'après cet **`PhotoSite`** exemple, et ajoutez une couche PHP App Server. Vous pouvez utiliser les valeurs par défaut pour les deux. Pour plus d'informations, consultez Créer une pile et Création d'une OpsWorks couche .
2. Sur la page Layers, pour PHP App Server, choisissez Security, puis choisissez Edit.
3. Dans la section Profil de couche, sélectionnez le nom du profil d'instance que vous avez enregistré précédemment, après le lancement de la AppServer AWS CloudFormation pile. Ce sera quelque chose comme`AppServer-AppServerInstanceProfile-1Q3KD0DNMGB90`. AWS OpsWorks Stacks attribue ce profil à toutes les EC2 instances Amazon de la couche, ce qui autorise les applications exécutées sur les instances de la couche à accéder à votre compartiment Amazon S3.



4. Ajoutez une instance à la couche PHP App Server et démarrez-la. Pour plus d'informations sur la façon d'ajouter et de démarrer des instances, consultez Ajout d'une instance à une couche.
5. Ajoutez une couche MySQL à la pile, ajoutez une instance et démarrez-la. Vous pouvez utiliser les paramètres par défaut pour la couche et pour l'instance. En particulier, l'instance MySQL n'a pas besoin d'accéder au compartiment Amazon S3. Elle peut donc utiliser le profil d'instance AWS OpsWorks Stacks standard, sélectionné par défaut.

Étape 3 : Créer et déployer un livre de recettes personnalisé

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

La pile n'est pas encore tout à fait prête :

- Votre application a besoin de certaines informations pour accéder au serveur de base de données MySQL et au compartiment Amazon S3, telles que le nom d'hôte de la base de données et le nom du compartiment Amazon S3.
- Vous devez configurer une base de données dans le serveur de base de données MySQL et créer une table pour contenir les métadonnées des photos.

Vous pouvez gérer ces tâches manuellement, mais une meilleure approche consiste à implémenter la recette Chef et à laisser AWS OpsWorks Stacks exécuter la recette automatiquement sur les instances appropriées. Les recettes Chef sont des applications Ruby spécialisées que AWS OpsWorks Stacks utilise pour effectuer des tâches sur des instances telles que l'installation de packages ou la création de fichiers de configuration. Elles sont rassemblées dans un livre de recettes, qui peut contenir plusieurs recettes et fichiers apparentés, tels que les modèles des fichiers de configuration. Le livre de recettes est placé dans un référentiel tel que GitHub, et doit avoir une structure de répertoire standard. Si vous n'avez pas encore un référentiel de livres de recettes personnalisé, consultez Référentiels de livres de recettes pour plus d'informations sur la façon d'en configurer un.

Dans cet exemple, le livre de recettes a été implémenté pour vous et est stocké dans un GitHub référentiel public. Le livre contient deux recettes, `appsetup.rb` et `dbsetup.rb`, et un fichier modèle, `db-connect.php.erb`.

La `appsetup.rb` recette crée un fichier de configuration qui contient les informations dont l'application a besoin pour accéder à la base de données et au compartiment Amazon S3. Il s'agit essentiellement d'une version légèrement modifiée de la recette `appsetup.rb` décrite dans Connecter l'application à la base de données. La différence principale est que les variables sont transmises au modèle, lesquelles représentent les informations d'accès.

Les quatre premiers attributs définissent les paramètres de connexion à la base de données et sont automatiquement définis par AWS OpsWorks Stacks lorsque vous créez l'instance MySQL.

Il existe deux différences entre ces variables et celles de la recette originale :

- Comme la recette originale, la variable `table` représente le nom de la table de base de données qui est créée par `dbsetup.rb` et elle est définie sur la valeur d'un attribut lui-même défini dans le fichier des attributs du livre de recettes.

  Cependant, l'attribut a un autre nom : `[:photoapp][:dbtable]`.

- La `s3bucket` variable est spécifique à cet exemple et est définie sur la valeur d'un attribut qui représente le nom du compartiment Amazon S3,`[:photobucket]`.

  `[:photobucket]` est défini en utilisant JSON personnalisé, comme décrit plus tard. Pour plus d'informations sur les attributs, consultez Attributs.

Pour plus d'informations sur les attributs, consultez Attributs.

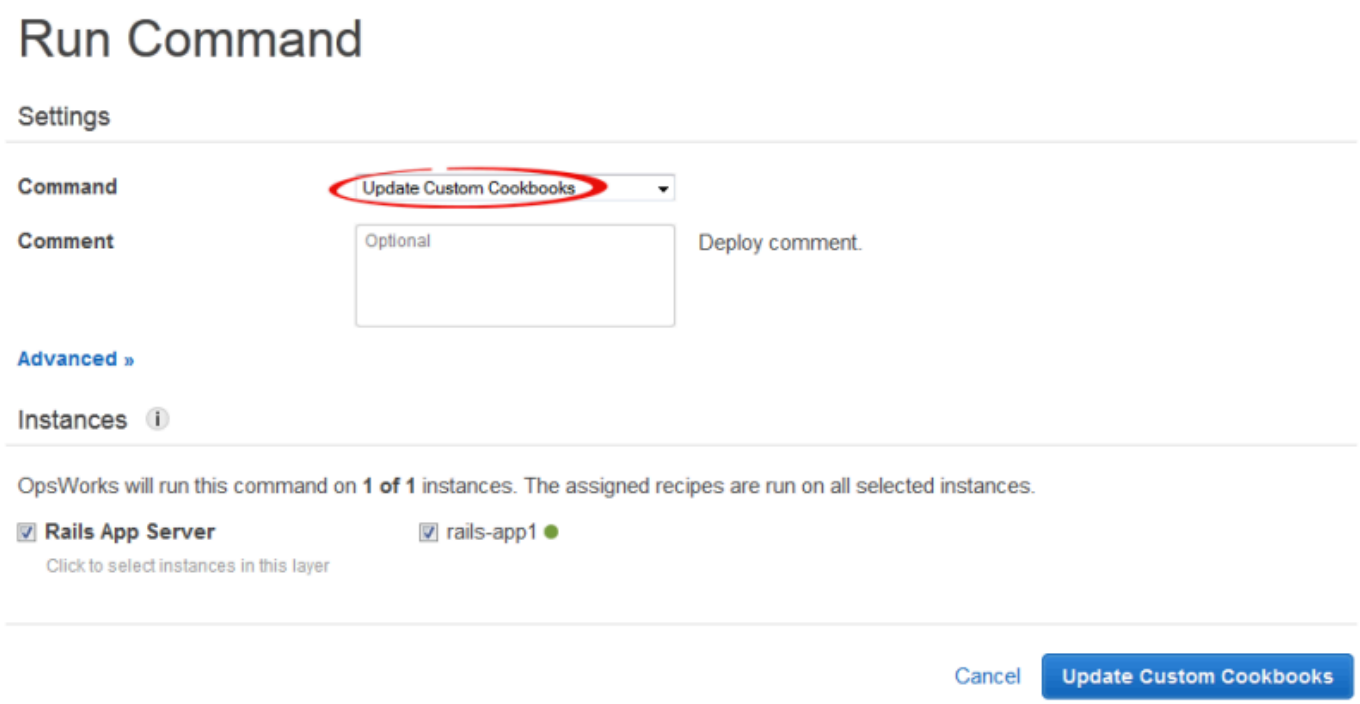
La recette `dbsetup.rb` installe une table de base de données pour contenir les métadonnées de chaque photo. Il s'agit d'une version légèrement modifiée de la recette `dbsetup.rb` décrite dans Configurer la base de données ; consultez la rubrique pour une description détaillée.

La seule différence entre cet exemple et la recette originale réside dans le schéma de base de données, qui comporte trois colonnes contenant l'identifiant, l'URL et la légende de chaque photo stockée dans le compartiment Amazon S3.

Les recettes étant déjà implémentées, il vous suffit de déployer le livre de recettes photoapp dans le cache du livre de recettes de chaque instance. AWS OpsWorks Stacks exécute ensuite les recettes mises en cache lorsque l'événement de cycle de vie approprié se produit, comme décrit plus loin.

Pour déployer le livre de recettes photoapp

1. Sur la page AWS OpsWorks Stacks Stack, choisissez Stack Settings, puis sélectionnez Modifier.
2. Dans la section Gestion de la configuration :
   - Définissez Utiliser les livres de recettes Chef personnalisés sur Oui.
   - Définissez Type de référentiel sur Git.
   - Définissez URL du référentiel sur **`git://github.com/amazonwebservices/opsworks-example-cookbooks.git`**.
3. Sur la page Pile, choisissez Exécuter la commande, sélectionnez la commande de pile Mettre à jour les livres de recettes personnalisés, puis choisissez Mettre à jour les livres de recettes personnalisés pour installer le nouveau livre de recettes dans les caches de livre de recettes de l'instance.



Étape 4 : Attribuer les recettes aux LifeCycle événements

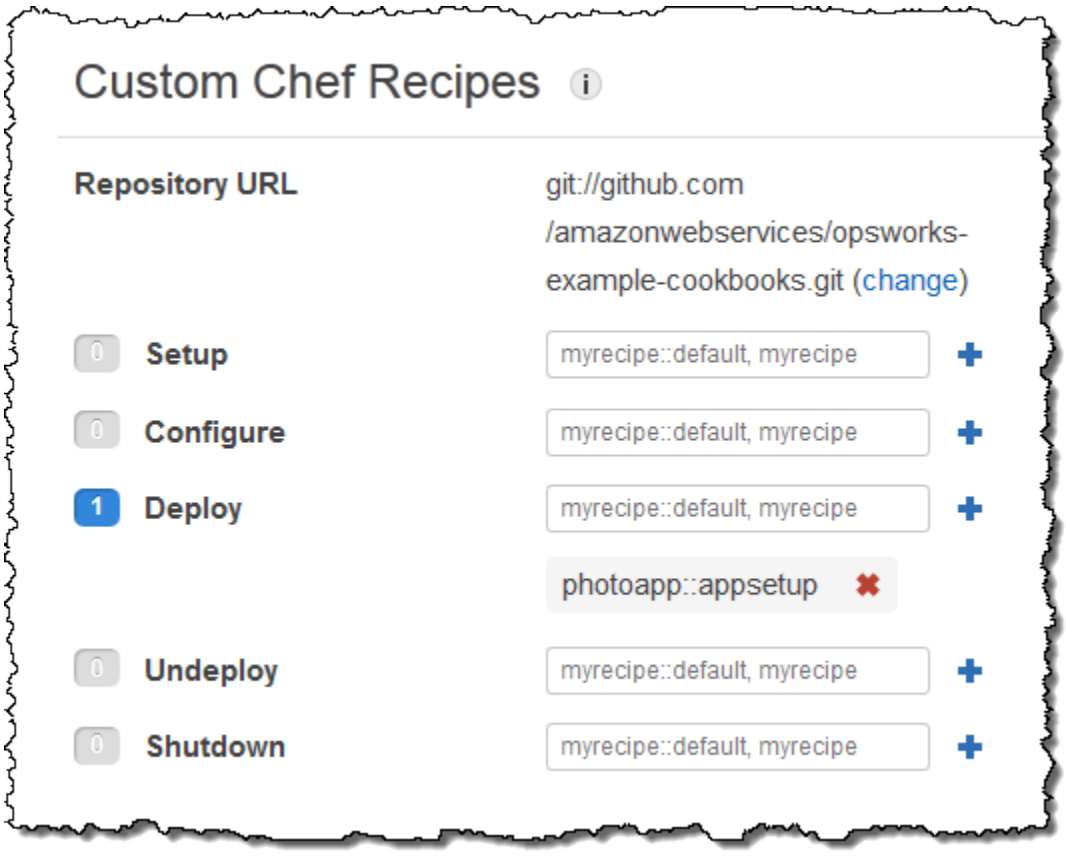
> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Vous pouvez exécuter des recettes personnalisées manuellement, mais la meilleure approche consiste généralement à faire en sorte que AWS OpsWorks Stacks les exécute automatiquement. Chaque couche possède un ensemble de recettes intégrées attribuées à chacun des cinq événements du cycle de vie (installation, configuration, déploiement, dédéploiement et arrêt). Chaque fois qu'un événement se produit sur une instance, AWS OpsWorks Stacks exécute les recettes associées pour chacune des couches de l'instance, qui gèrent les tâches requises. Par exemple, lorsqu'une instance termine le démarrage, AWS OpsWorks Stacks déclenche un événement de configuration pour exécuter les recettes de configuration, qui gèrent généralement des tâches telles que l'installation et la configuration de packages.

Vous pouvez demander à AWS OpsWorks Stacks d'exécuter des recettes personnalisées sur les instances d'une couche en affectant chaque recette à l'événement du cycle de vie approprié. AWS OpsWorks Stacks exécutera toutes les recettes personnalisées une fois les recettes intégrées à la couche terminées. Pour cet exemple, attribuez-le `appsetup.rb` à l'événement Deploy de la couche PHP App Server et `dbsetup.rb` à l'événement Deploy de la couche MySQL. AWS OpsWorks Stacks exécutera ensuite les recettes sur les instances de la couche associée au démarrage, une fois les recettes de configuration intégrées terminées, et chaque fois que vous déployez une application, une fois les recettes de déploiement créées terminées. Pour de plus amples informations, veuillez consulter Exécution automatique des recettes.

Pour attribuer des recettes personnalisées à l'événement Deploy de la couche

1. Sur la page AWS OpsWorks Stacks Layers, pour le serveur d'applications PHP, choisissez Recipes, puis Edit.
2. Sous Recettes Chef personnalisées, ajoutez le nom de la recette à l'événement Deploy (Déployer) et choisissez +. Le nom doit être au format Chef *`cookbookname`*`::`*`recipename`*, où *`recipename`* n'inclut pas l'extension `.rb`. Pour cet exemple, vous saisissez `photoapp::appsetup`. Ensuite, choisissez Enregistrer pour mettre à jour la configuration de la couche.



3. Sur la page Couches, choisissez Modifier dans la colonne Actions de la couche MySQL.
4. Ajoutez `photoapp::dbsetup` à l'événement Deploy de la couche et enregistrez la nouvelle configuration.

Étape 5 : Ajouter les informations d'accès aux attributs de configuration et de déploiement de la pile

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

La `appsetup.rb` recette dépend des données issues de la configuration de la pile AWS OpsWorks Stacks et des attributs de déploiement, qui sont installés sur chaque instance et contiennent des informations détaillées sur la pile et les applications déployées. Les attributs `deploy` de l'objet ont la structure suivante, affichée pour plus de commodité au format JSON :

```
{
   ...
  "deploy": {
    "app1": {
      "application" : "short_name",
      ...
    }
    "app2": {
      ...
    }
    ...
  }
}
```

Le nœud Deploy contient un attribut pour chaque application déployée, nommé d'après le nom court de l'application. Chaque attribut d'application contient un ensemble d'attributs qui définissent la

configuration de l'application, tels que la racine du document et le type d'application. Pour obtenir la liste des attributs `deploy`, consultez Attributs deploy. Vous pouvez représenter les valeurs des attributs de déploiement et de configuration de la pile dans vos recettes en utilisant la syntaxe d'attribut Chef. Par exemple, `[:deploy][:app1][:application]` représente le nom court de l'application app1.

Les recettes personnalisées dépendent de plusieurs attributs de configuration et de déploiement de la pile qui représentent les informations d'accès à la base de données et à Amazon S3 :

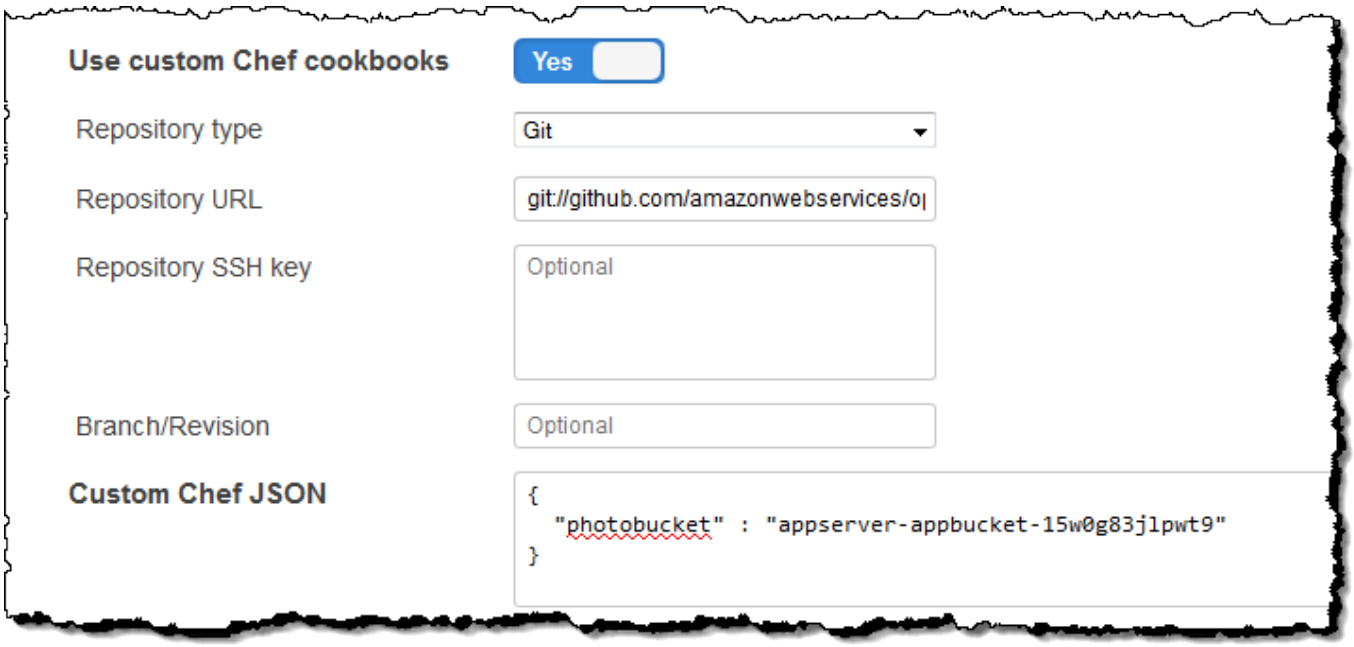
- Les attributs de connexion à la base de données, tels que`[:deploy][:database][:host]`, sont définis par AWS OpsWorks Stacks lors de la création de la couche MySQL.
- L'attribut de nom de table, `[:photoapp][:dbtable]`, est défini dans le fichier des attributs du livre de recettes personnalisé, et a pour valeur `foto`.
- Vous devez définir l'attribut de nom de compartiment, `[:photobucket]`, en utilisant le JSON personnalisé pour ajouter l'attribut aux attributs de configuration et de déploiement de la pile.

Pour définir l'attribut de nom de compartiment Amazon S3

1. Sur la page AWS OpsWorks Stacks Stack, choisissez Stack Settings, puis Modifier.
2. Dans la section Configuration Management, ajoutez les informations d'accès à la zone Custom Chef JSON. Elles doivent se présenter comme suit :

   ```
   {
     "photobucket" : "yourbucketname"
   }
   ```

   *yourbucketname*Remplacez-le par le nom du bucket dans lequel vous avez enregistréÉtape 1 : Créer un compartiment Amazon S3.



AWS OpsWorks Stacks fusionne le JSON personnalisé dans les attributs de configuration et de déploiement de la pile avant de les installer sur les instances de la pile ; il `appsetup.rb` peut ensuite obtenir le nom du bucket à partir de l'attribut. `[:photobucket]` Si vous voulez modifier le compartiment, vous n'avez pas besoin de toucher la recette ; vous pouvez simplement remplacer l'attribut pour fournir un nouveau nom de compartiment.

Étape 6 : Déploiement et exécution PhotoApp

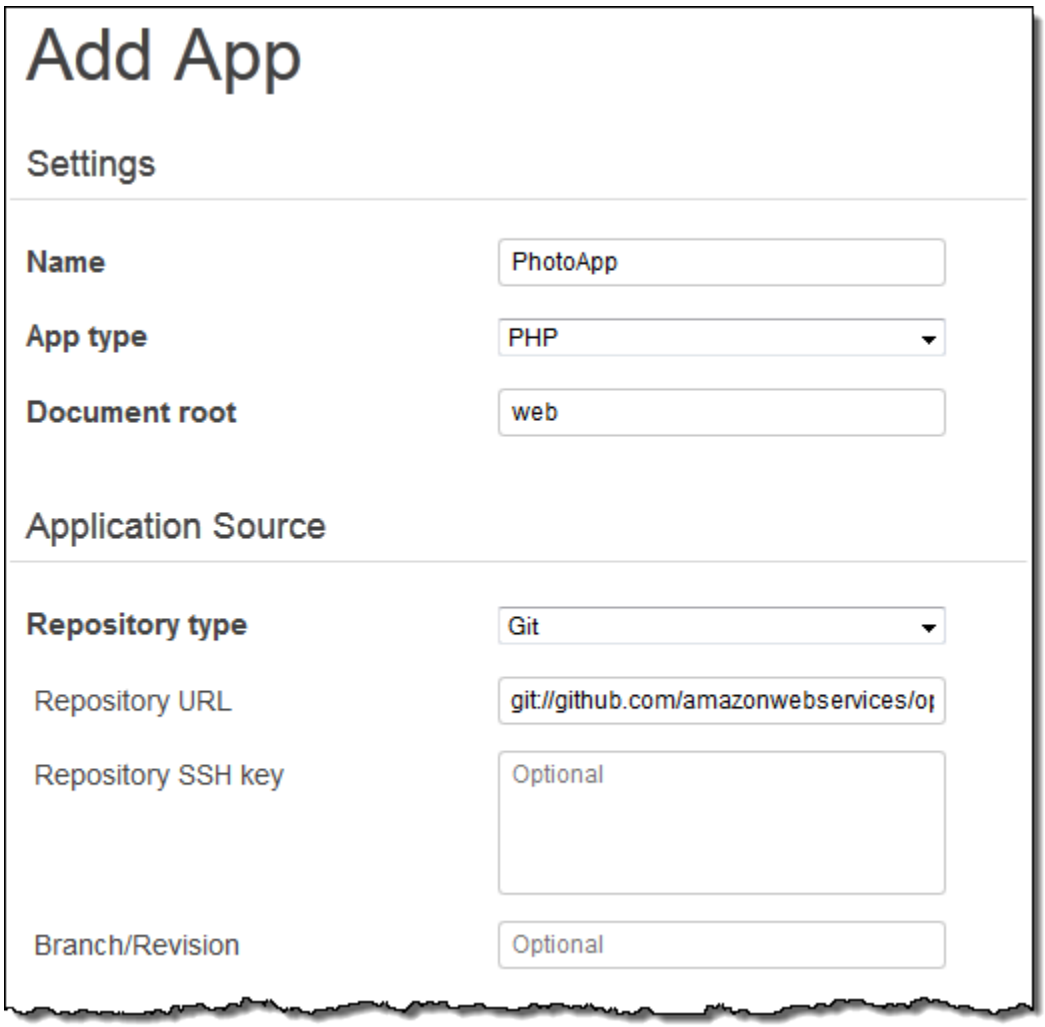
> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Dans cet exemple, l'application a également été implémentée pour vous et est stockée dans un GitHub référentiel public. Vous devez simplement ajouter l'application à la pile, la déployer sur les serveurs d'applications et l'exécuter.

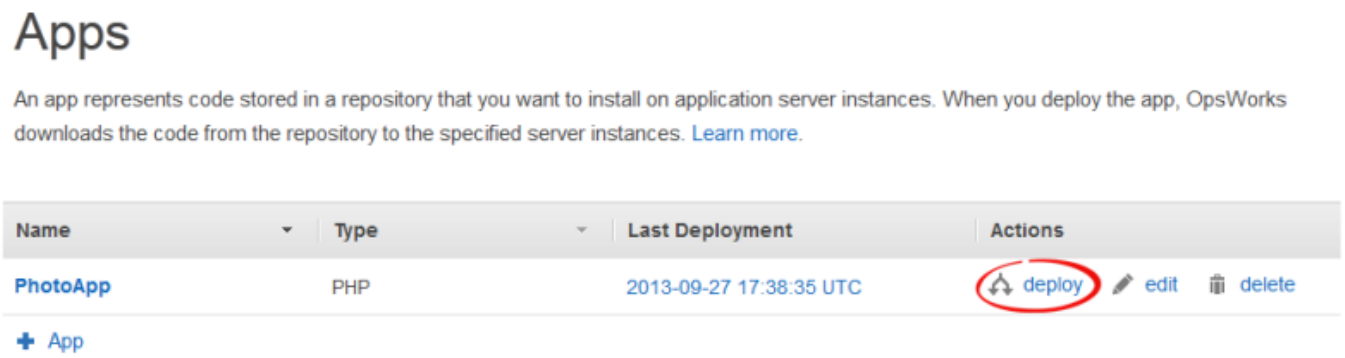
Pour ajouter l'application à la pile et la déployer sur les serveurs d'applications

1. Ouvrez la page Apps (Applications) et choisissez Add an app (Ajouter une application).
2. Sur la page Add App, procédez de la façon suivante :

   - Définissez Nom sur **`PhotoApp`**.
   - Définissez Type d'application sur PHP.
   - Définissez Racine du document sur **`web`**.
   - Définissez Type de référentiel sur Git.
   - Définissez URL du référentiel sur **`git://github.com/awslabs/opsworks-demo-php-photo-share-app.git`**.
   - Choisissez Ajouter une application pour accepter les valeurs par défaut pour les autres paramètres.

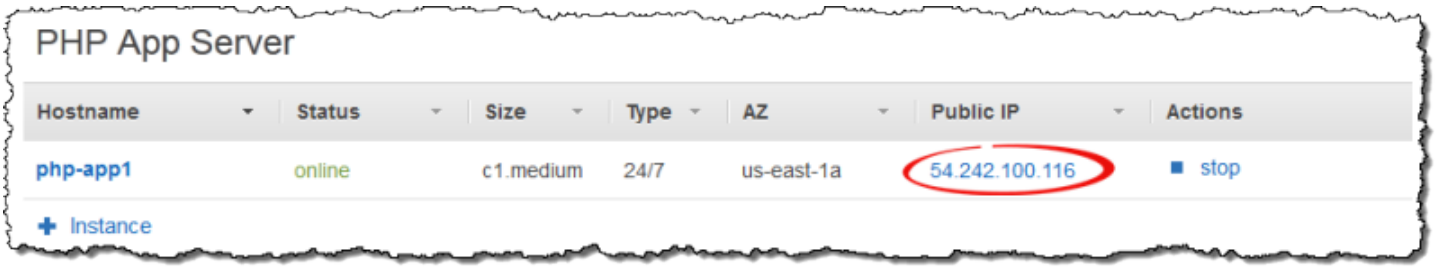


3. Sur la page Applications, choisissez déployer dans la colonne Actions de PhotoApp l'application.



4. Acceptez les valeurs par défaut et choisissez Déployer pour déployer l'application sur le serveur.

Pour exécuter PhotoApp, rendez-vous sur la page Instances et choisissez l'adresse IP publique de l'instance PHP App Server.



Vous devez voir l'interface utilisateur suivante. Choisissez Ajouter une photo pour stocker une photo dans le compartiment Amazon S3 et les métadonnées dans le magasin de données principal.



Utilisation AWS CodePipeline avec AWS OpsWorks Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS CodePipelinevous permet de créer des pipelines de livraison continus qui suivent les modifications de code provenant de sources telles qu' CodeCommitAmazon Simple Storage Service (Amazon S3) ou. GitHub Vous pouvez l'utiliser CodePipeline pour automatiser la publication de vos livres de recettes Chef et de votre code d'application vers AWS OpsWorks Stacks, sur les stacks Chef 11.10, Chef 12 et Chef 12.2. Les exemples de cette section décrivent comment créer et utiliser un pipeline simple CodePipeline comme outil de déploiement pour le code que vous exécutez sur des couches AWS OpsWorks Stacks.

Note

CodePipeline et l'intégration de AWS OpsWorks Stacks n'est pas prise en charge pour le déploiement sur Chef 11.4 et les anciennes piles.

Rubriques

- AWS CodePipeline avec AWS OpsWorks Stacks - Chef 12 Stacks
- AWS CodePipeline avec AWS OpsWorks Stacks - Chef 11 Stacks

AWS CodePipeline avec AWS OpsWorks Stacks - Chef 12 Stacks

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS CodePipelinevous permet de créer des pipelines de livraison continus qui suivent les modifications de code provenant de sources telles qu' CodeCommitAmazon Simple Storage Service (Amazon S3) ou. GitHub L'exemple présenté dans cette rubrique décrit comment créer et utiliser un pipeline simple CodePipeline comme outil de déploiement pour le code que vous exécutez sur

des couches AWS OpsWorks Stacks. Dans cet exemple, vous créez un pipeline pour une simple application Node.js, puis vous demandez à AWS OpsWorks Stacks d'exécuter l'application sur toutes les instances d'une couche d'une pile Chef 12 (dans ce cas, une seule instance).

> **Note**
>
> Cette rubrique indique comment utiliser un pipeline pour exécuter et mettre à jour une application sur une pile Chef 12. Pour plus d'informations sur l'utilisation d'un pipeline pour exécuter et mettre à jour une application sur une pile Chef 11.10, consultezAWS CodePipeline avec AWS OpsWorks Stacks - Chef 11 Stacks. Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Rubriques

- Prérequis
- Autres scénarios pris en charge
- Étape 1 : créer une pile, une couche et une instance dans AWS OpsWorks Stacks
- Étape 2 : Configurer votre pile et la couche de manière à utiliser les livres de recettes personnalisés
- Étape 3 : Charger le code de l'application dans un compartiment Amazon S3
- Étape 4 : Ajout de votre application à AWS OpsWorks Stacks
- Étape 5 : Créer un pipeline dans CodePipeline
- Étape 6 : Vérification du déploiement de l'application dans AWS OpsWorks Stacks
- Étape 7 (facultatif) : Mettre à jour le code d'application pour voir CodePipeline redéployer votre application automatiquement
- Étape 8 (facultatif) : Nettoyer les ressources

Prérequis

Avant de commencer cette procédure, veillez à disposer des autorisations d'administrateur pour exécuter les tâches suivantes. Vous pouvez être membre d'un groupe auquel la AdministratorAccesspolitique est appliquée, ou vous pouvez être membre d'un groupe doté des

autorisations et des politiques indiquées dans le tableau suivant. Pour des raisons de sécurité, vous devez appartenir à un groupe autorisé à effectuer les tâches suivantes, au lieu d'attribuer les autorisations requises à des utilisateurs individuels.

Pour plus d'informations sur la création d'un groupe de sécurité dans IAM et l'attribution d'autorisations au groupe, consultez la section Création de groupes d'utilisateurs IAM. Pour plus d'informations sur la gestion des autorisations AWS OpsWorks Stacks, consultez Bonnes pratiques : gestion des autorisations.

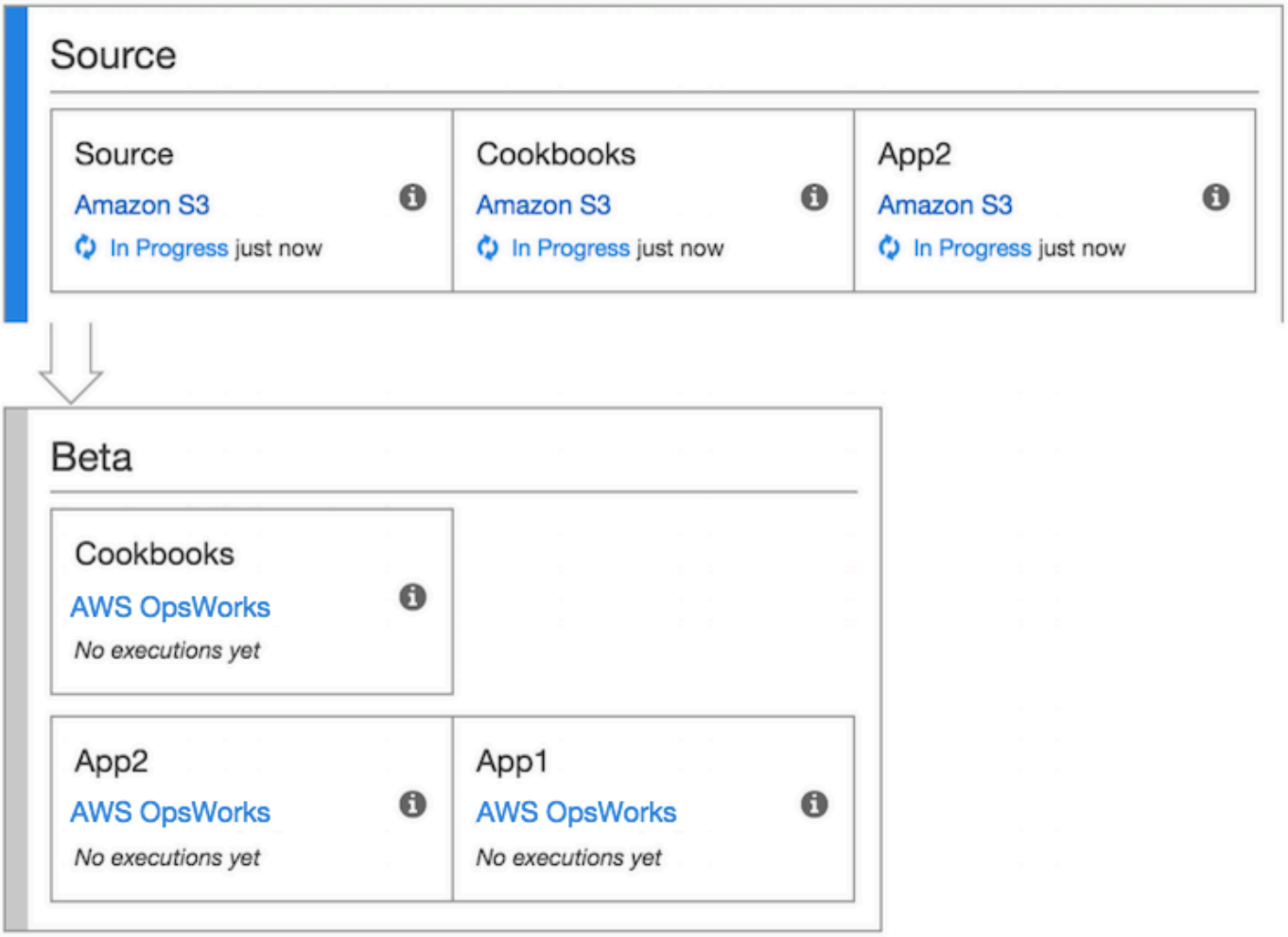
Autorisations	Stratégie recommandée pour attacher à un groupe
Créez et modifiez des piles, des couches et des instances dans AWS OpsWorks Stacks.	AWSOpsWorks_FullAccess
Créez, modifiez et exécutez les modèles dans AWS CloudFormation.	AmazonCloudFormationFullAccess
Créez, modifiez et accédez aux compartiments Amazon S3.	Amazon S3 FullAccess
Créez, modifiez et exécutez des pipelines CodePipeline, en particulier des pipelines qui utilisent AWS OpsWorks Stacks comme fournisseur.	AWSCodePipeline_FullAccess

Vous devez également disposer d'une paire de EC2 clés Amazon. Dans cette procédure pas à pas, vous serez invité à fournir le nom de cette paire de clés lorsque vous exécuterez le AWS CloudFormation modèle qui crée la pile d'échantillons, la couche et l'instance. Pour plus d'informations sur l'obtention d'une paire de clés dans la EC2 console Amazon, consultez la section Créer une paire de clés dans la EC2 documentation Amazon. La paire de clés doit résider dans la région USA Est (Virginie du Nord). Vous pouvez utiliser une paire de clés existante, si vous en avez déjà une dans cette région.

Autres scénarios pris en charge

Cette procédure crée un simple pipeline qui inclut une étape Source et une étape Deploy (Déployer). Cependant, vous pouvez créer des pipelines plus complexes qui utilisent AWS OpsWorks Stacks comme fournisseur. Les exemples suivants concernent les pipelines et les scénarios pris en charge :

- Vous pouvez modifier un pipeline pour ajouter un livre de recettes Chef à l'étape Source et une cible associée pour les livres de recettes mis à jour à l'étape Deploy (Déployer). Dans ce cas, vous ajoutez une action Deploy (Déployer) qui déclenche la mise à jour de vos livres de recettes lorsque vous apportez des modifications à la source. Le livre de recettes mis à jour est déployé avant votre application.
- Vous pouvez créer un pipeline complexe, avec des livres de recettes personnalisés et plusieurs applications, et le déployer sur une AWS OpsWorks pile Stacks. Le pipeline suit les modifications apportées à la fois aux sources de l'application et aux sources des livres de recettes, et redéploie lorsque vous avez apporté les modifications. Vous trouverez ci-dessous un exemple de pipeline complexe et similaire:



Pour plus d'informations sur l'utilisation CodePipeline, veuillez consulter le Guide de CodePipeline l'utilisateur.

Étape 1 : créer une pile, une couche et une instance dans AWS OpsWorks Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Pour utiliser AWS OpsWorks Stacks comme fournisseur de déploiement pour un pipeline, vous devez d'abord disposer d'une pile, d'une couche et d'au moins une instance dans la couche. Bien que vous puissiez créer une pile dans AWS OpsWorks Stacks en suivant les instructions de Getting Started with Linux Stacks ou Getting Started with Windows Stacks, pour gagner du temps, cet exemple utilise un AWS CloudFormation modèle pour créer une pile, une couche et une instance Chef 12 basées sur Linux. L'instance créée par ce modèle exécute Amazon Linux 2016.03 et possède `c3.large` comme type d'instance. Bien que le modèle ne configure pas votre pile pour utiliser les livres de recettes personnalisés, vous le ferez ultérieurement dans la procédure pas à pas.

> **Important**
>
> Le AWS CloudFormation modèle doit être stocké et exécuté dans la même région que le compartiment Amazon S3 dans lequel vous téléchargerez ultérieurement votre application et dans la même région dans laquelle vous créerez ultérieurement votre pipeline CodePipeline. Pour le moment, CodePipeline prend uniquement en charge le fournisseur AWS OpsWorks Stacks dans la région USA Est (Virginie du Nord) (us-east-1). Toutes les ressources de cette procédure doivent être créées dans la région USA Est (Virginie du Nord).
>
> Si la création de la pile échoue, il se peut que vous approchiez du nombre maximal de rôles IAM pour votre compte. La création de piles peut également échouer si votre compte ne peut pas lancer les instances avec un type d'instance `c3.large`. Par exemple, si vous utilisez le niveau AWS gratuit, vous pouvez recevoir un message d'erreur tel que`Root device type: must be included in EBS`. Si votre compte comporte des limites quant aux types d'instances que vous êtes autorisé à créer, telles que les limites imposées par le niveau AWS gratuit, essayez de remplacer la valeur du `InstanceType` paramètre dans le bloc d'instances du modèle par un type d'instance que votre compte peut utiliser.

Pour créer une pile, une couche et une instance à l'aide de AWS CloudFormation

1. Copiez le AWS CloudFormation modèle suivant dans un nouveau document en texte brut. Enregistrez le fichier à un emplacement approprié sur votre ordinateur local et nommez-le NewOpsWorksStack.template, ou un autre nom qui vous convient.

```
{
  "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": [
```

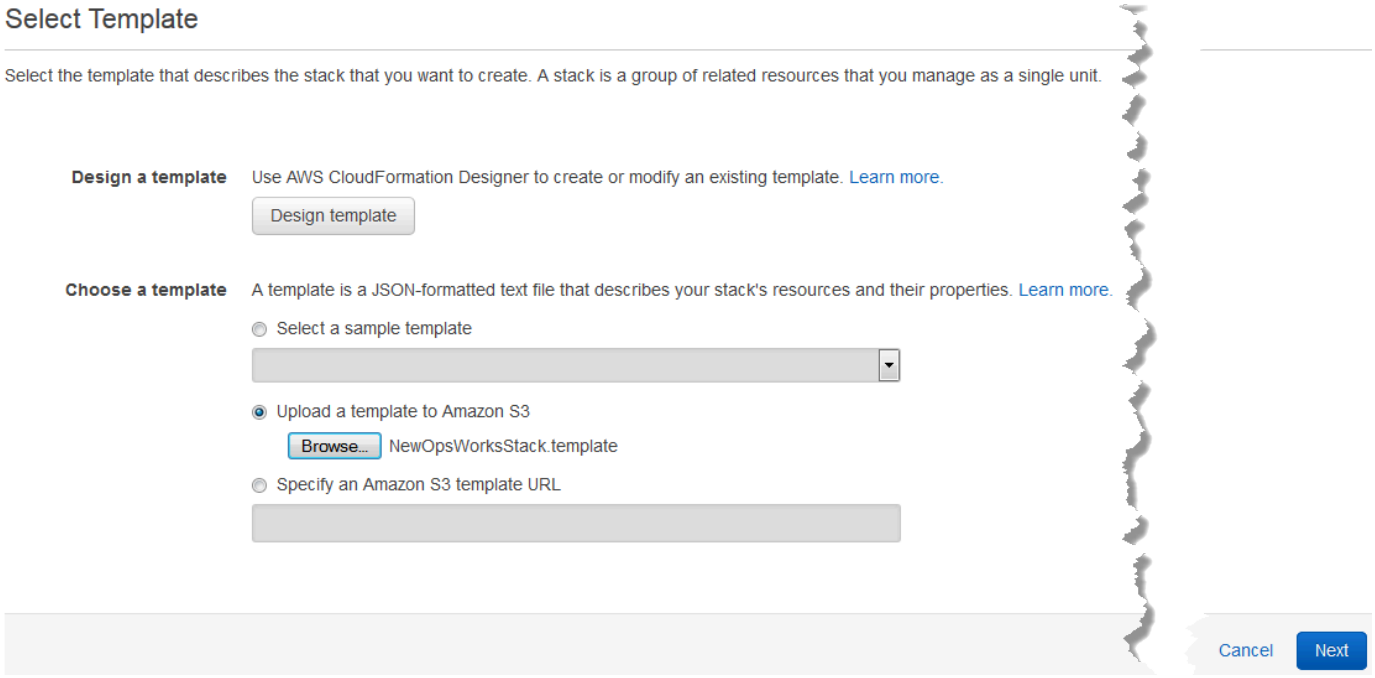
```
        "MyStack",
        "CPOpsDeploySecGroup"
      ]
    },
    "OpsWorksServiceRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "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. Connectez-vous à la AWS CloudFormation console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudformation.
3. Sur la page d' AWS CloudFormation accueil, choisissez Create stack.
4. Sur la page Sélectionner un modèle, dans la zone Choisir un modèle, choisissez Télécharger un modèle sur Amazon S3, puis Parcourir.
5. Accédez au AWS CloudFormation modèle que vous avez enregistré à l'étape 1, puis choisissez Ouvrir. Sur la page Select Template, choisissez Next.



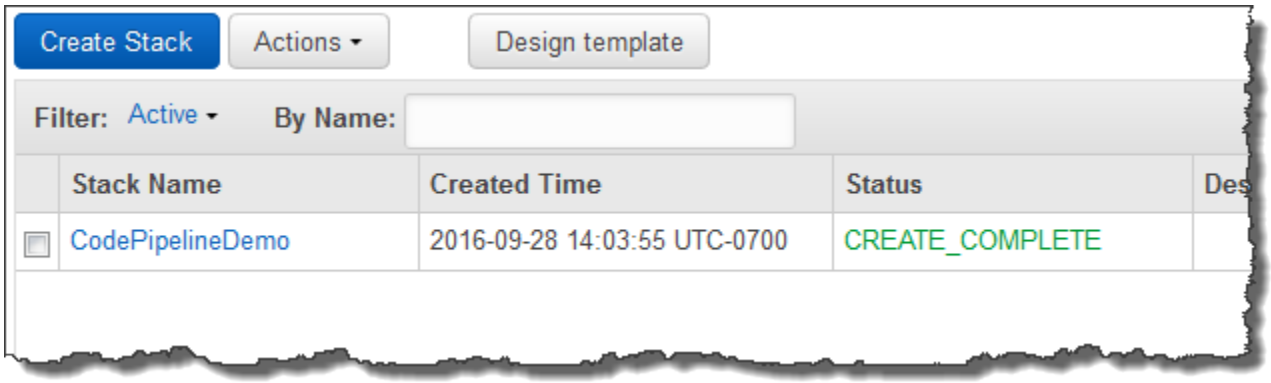
6. Sur la page Spécifier les détails, nommez la pile CodePipelineDemoou tout autre nom de pile propre à votre compte. Si vous choisissez un autre nom pour votre pile, modifiez le nom de la pile tout au long de cette procédure pas à pas.
7. Dans la zone Paramètres, indiquez le nom d'une paire de EC2 clés que vous souhaitez utiliser pour accéder à votre instance AWS OpsWorks Stacks une fois celle-ci créée. Choisissez Suivant.
8. Dans la page Options, choisissez Suivant. (Les paramètres de cette page ne sont pas obligatoires pour cette procédure pas à pas.)

9. Le AWS CloudFormation modèle que vous utilisez dans cette procédure pas à pas crée des rôles IAM, un profil d'instance et une instance.

   > **Important**
   >
   > Avant de choisir Créer, choisissez Coût pour estimer les frais que vous pourriez encourir AWS pour créer des ressources avec ce modèle.

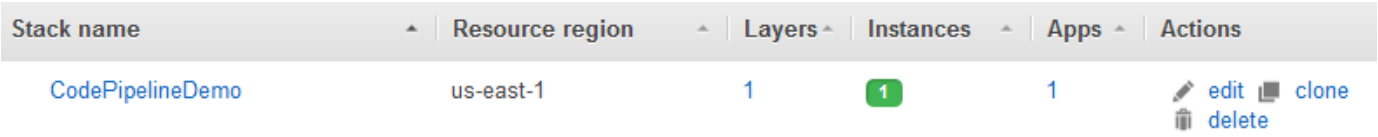
   Si la création de ressources IAM est acceptable, cochez la case Je reconnais que ce modèle peut AWS CloudFormation entraîner la création de ressources IAM, puis choisissez Créer.
   Si la création de ressources IAM n'est pas acceptable, vous ne pouvez pas poursuivre cette procédure.

10. Le AWS CloudFormation tableau de bord vous permet de visualiser la progression de la création de la pile. Avant de passer à l'étape suivante, patientez jusqu'à ce que CREATE_COMPLETE s'affiche dans la colonne Statut.



Pour vérifier la création d'une pile dans AWS OpsWorks Stacks

1. Ouvrez la AWS OpsWorks console à l'adresse https://console.aws.amazon.com/opsworks/.
2. Sur le tableau de bord AWS OpsWorks Stacks, visualisez la pile que vous avez créée.



3. Ouvrez la pile, puis affichez la couche et l'instance. Notez que la couche et l'instance ont été créées avec les noms et autres métadonnées fournis dans le AWS CloudFormation modèle.

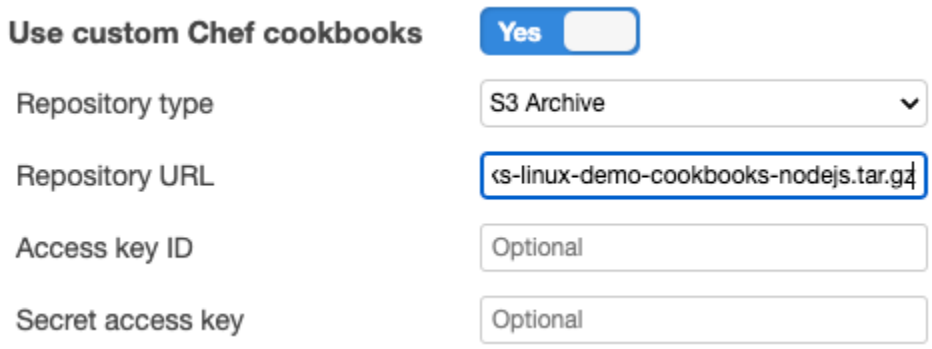
Vous êtes prêt à configurer votre pile et la couche de manière à utiliser les livres de recettes et les recettes personnalisés de Chef.

Étape 2 : Configurer votre pile et la couche de manière à utiliser les livres de recettes personnalisés

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les piles Chef 12 dans AWS OpsWorks Stacks nécessitent vos propres livres de recettes ou ceux créés par la communauté pour créer des couches d'application personnalisées. Pour cette procédure, vous pouvez pointer vers un référentiel qui contient un ensemble de livres de recettes Chef et des recettes Chef. Ces recettes installent le package Node.js et ses dépendances sur votre instance. Vous utiliserez d'autres recettes Chef pour déployer l'application Node.js que vous préparerez dans Étape 4 : Ajout de votre application à AWS OpsWorks Stacks. La recette Chef que vous spécifiez à cette étape s'exécute chaque fois qu'une nouvelle version de votre application est déployée par CodePipeline.

1. Dans la console AWS OpsWorks Stacks, ouvrez la pile que vous avez créée dansÉtape 1 : créer une pile, une couche et une instance dans AWS OpsWorks Stacks. Choisissez Paramètres de pile, puis Modifier.
2. Définissez Utiliser les livres de recettes Chef personnalisés sur Oui. Ceci permet d'afficher les paramètres personnalisés du livre de recettes associé.
3. Dans la liste déroulante Type de référentiel, choisissez Archive S3. Pour fonctionner avec les deux AWS OpsWorks, CodePipeline la source de votre livre de recettes doit être S3.
4. Pour URL du référentiel, indiquez **`https://s3.amazonaws.com/opsworks-demo-assets/opsworks-linux-demo-cookbooks-nodejs.tar.gz`**. Vos paramètres doivent ressembler à ce qui suit.



5. Choisissez Enregistrer.
6. Choisissez Layers dans le volet de navigation.
7. Choisissez Paramètres pour la couche créée dans Étape 1 : créer une pile, une couche et une instance dans AWS OpsWorks Stacks.
8. Dans l'onglet Paramètres généraux, vérifiez que le nom de la couche est Serveur d'applications Node.js et que le nom abrégé de la couche est app1. Choisissez Recettes.
9. Dans l'onglet Recettes, indiquez **`nodejs_demo`** comme recette à exécuter au cours de l'événement de cycle de vie Deploy (Déployer). Choisissez Enregistrer.
10. Dans l'onglet Sécurité, dans la liste déroulante Groupes de sécurité, sélectionnez le groupe de sécurité AWS- OpsWorks -Webapp.
11. Choisissez Enregistrer.

Étape 3 : Charger le code de l'application dans un compartiment Amazon S3

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

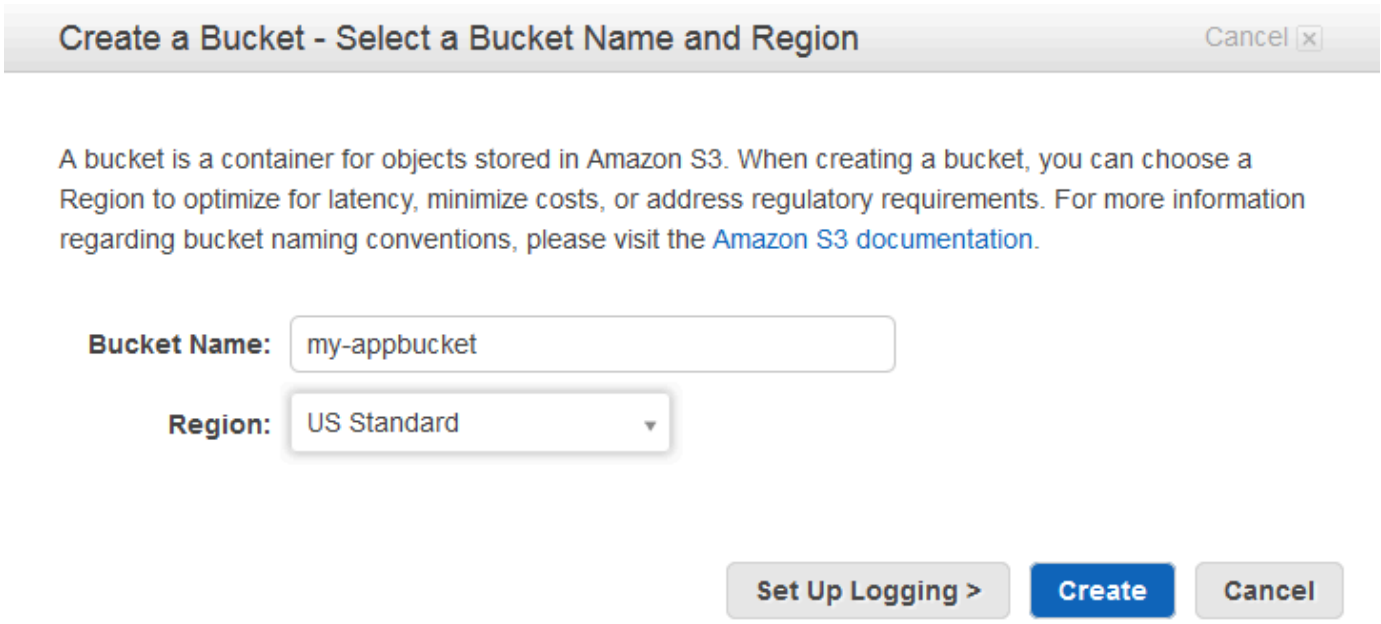
Comme vous devez fournir un lien vers le référentiel de votre code dans le cadre de l'installation du pipeline, préparez le référentiel avant de créer le pipeline. Dans cette procédure, vous chargez une application Node.js dans un compartiment Amazon S3.

Bien qu'il soit CodePipeline possible d'utiliser du code directement depuis GitHub ou CodeCommit en tant que source, cette procédure pas à pas explique comment utiliser un compartiment Amazon S3. Dans cette procédure pas à pas, vous allez télécharger l'exemple d'application Node.js dans votre propre compartiment Amazon S3 afin de pouvoir apporter des modifications à l'application. Le compartiment Amazon S3 que vous créez à cette étape permet de CodePipeline détecter les modifications apportées au code de l'application et de déployer automatiquement l'application modifiée. Si vous le souhaitez, vous pouvez utiliser un compartiment existant. Assurez-vous que le compartiment répond aux critères décrits dans la section Procédure pas à pas simple du pipeline (compartiment Amazon S3) de la CodePipeline documentation.

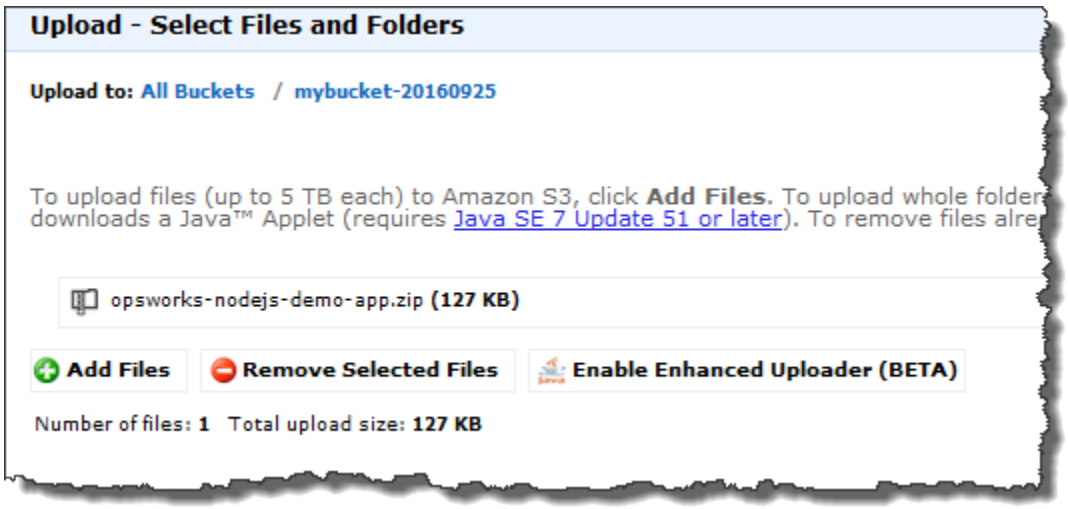
> **Important**
>
> Le compartiment Amazon S3 doit se trouver dans la même région que celle dans laquelle vous créerez votre pipeline. Pour le moment, CodePipeline prend uniquement en charge le fournisseur AWS OpsWorks Stacks dans la région USA Est (Virginie du Nord) (us-east-1). Toutes les ressources de cette procédure doivent être créées dans la région USA Est (Virginie du Nord). Le bucket doit également être versionné car il CodePipeline nécessite une source versionnée. Pour plus d'informations, consultez Utilisation de la gestion des versions.

Pour télécharger votre application dans un compartiment Amazon S3

1. Téléchargez le fichier ZIP de l'exemple AWS OpsWorks Stacks, l'application Node.js, et enregistrez-le à un emplacement pratique sur votre ordinateur local.
2. Ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
3. Choisissez Créer un compartiment.
4. Sur la page Créer un compartiment - Sélectionner un nom et une région de compartiment, pour Bucket Name (Nom du compartiment), saisissez un nom unique pour votre compartiment. Les noms de compartiment doivent être uniques dans tous les AWS comptes, et pas seulement dans votre propre compte. Cette procédure utilise le nom **`my-appbucket`**, mais vous pouvez utiliser `my-appbucket-`*`yearmonthday`* pour que votre nom de compartiment soit unique. Dans la liste déroulante Region, choisissez US Standard, puis Créer. USA Standard est l'équivalent de `us-east-1`.



5. Choisissez le compartiment créé à partir de la liste Tous les compartiments.
6. Sur la page du compartiment, choisissez Charger.
7. Sur la page Charger - Sélectionner les fichiers et dossiers, choisissez Ajouter des fichiers. Accédez au fichier ZIP enregistré à l'étape 1, choisissez Ouvrir, puis choisissez Commencer le chargement.



8. Une fois le chargement terminé, sélectionnez le fichier ZIP dans la liste des fichiers de votre compartiment, puis choisissez Propriétés.
9. Dans le volet Propriétés, copiez le lien vers votre fichier ZIP et notez le lien. Vous aurez besoin du nom de compartiment et de la partie nom de fichier ZIP du lien pour créer votre pipeline.

Étape 4 : Ajout de votre application à AWS OpsWorks Stacks

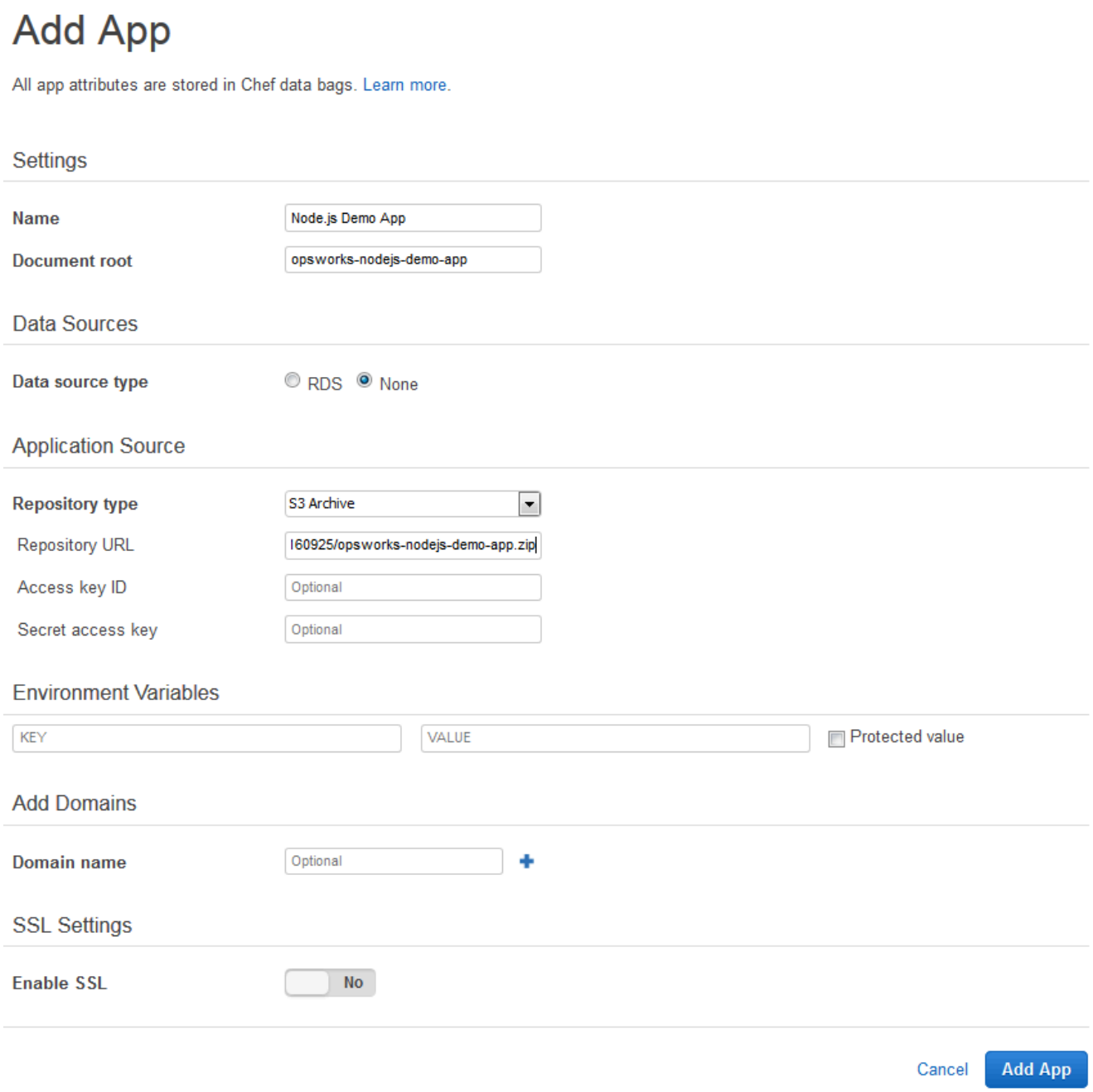
> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Avant de créer un pipeline CodePipeline, ajoutez l'application de test Node.js à AWS OpsWorks Stacks. Lorsque vous créez le pipeline, vous devez sélectionner l'application que vous avez ajoutée à AWS OpsWorks Stacks.

Préparez le lien vers le compartiment Amazon S3 de l'étape 9 de la procédure précédente. Vous aurez besoin du lien au compartiment dans lequel vous avez stocké votre application de test pour terminer cette procédure.

Pour ajouter une application à AWS OpsWorks Stacks

1. Dans la console AWS OpsWorks Stacks CodePipelineDemo, ouvrez et dans le volet de navigation, sélectionnez Apps.
2. Choisissez Add app (Ajouter une application).
3. Sur la page Add App (Ajouter une application), indiquez les informations suivantes :
   a. Spécifiez un nom pour votre application. Cette procédure utilise le nom `Node.js Demo App`.
   b. Pour Type de source de données, choisissez Aucun. Cette application ne requiert pas une source de données ou base de données externe.
   c. Dans la liste déroulante Type de référentiel, choisissez Archive S3.
   d. Dans la zone URL du référentiel, collez l'URL que vous avez copiée à l'étape 9 de Étape 3 : Charger le code de l'application dans un compartiment Amazon S3. Votre formulaire doit être similaire à ce qui suit:



4. Vous n'avez pas besoin de modifier d'autres paramètres du formulaire. Choisissez Add App (Ajouter une application).
5. Lorsque l'application Node.js Demo App s'affiche dans la liste de la page Applications, passez à la procédure suivante, Étape 5 : Créer un pipeline dans CodePipeline.

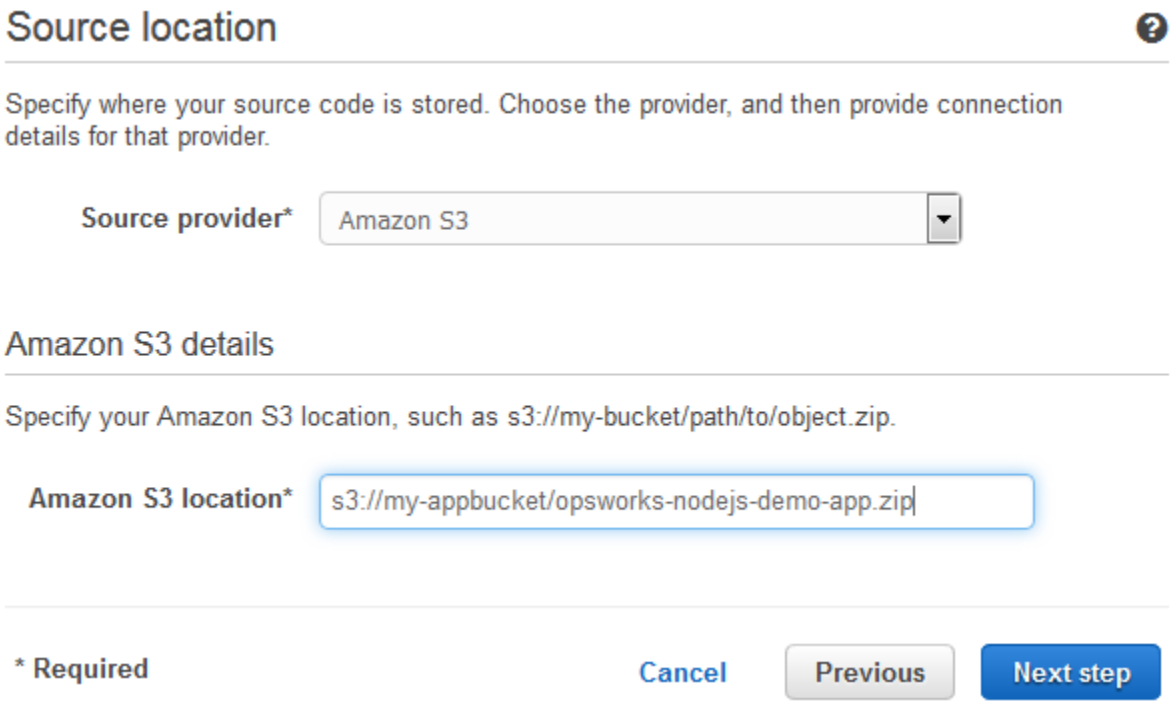
Étape 5 : Créer un pipeline dans CodePipeline

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

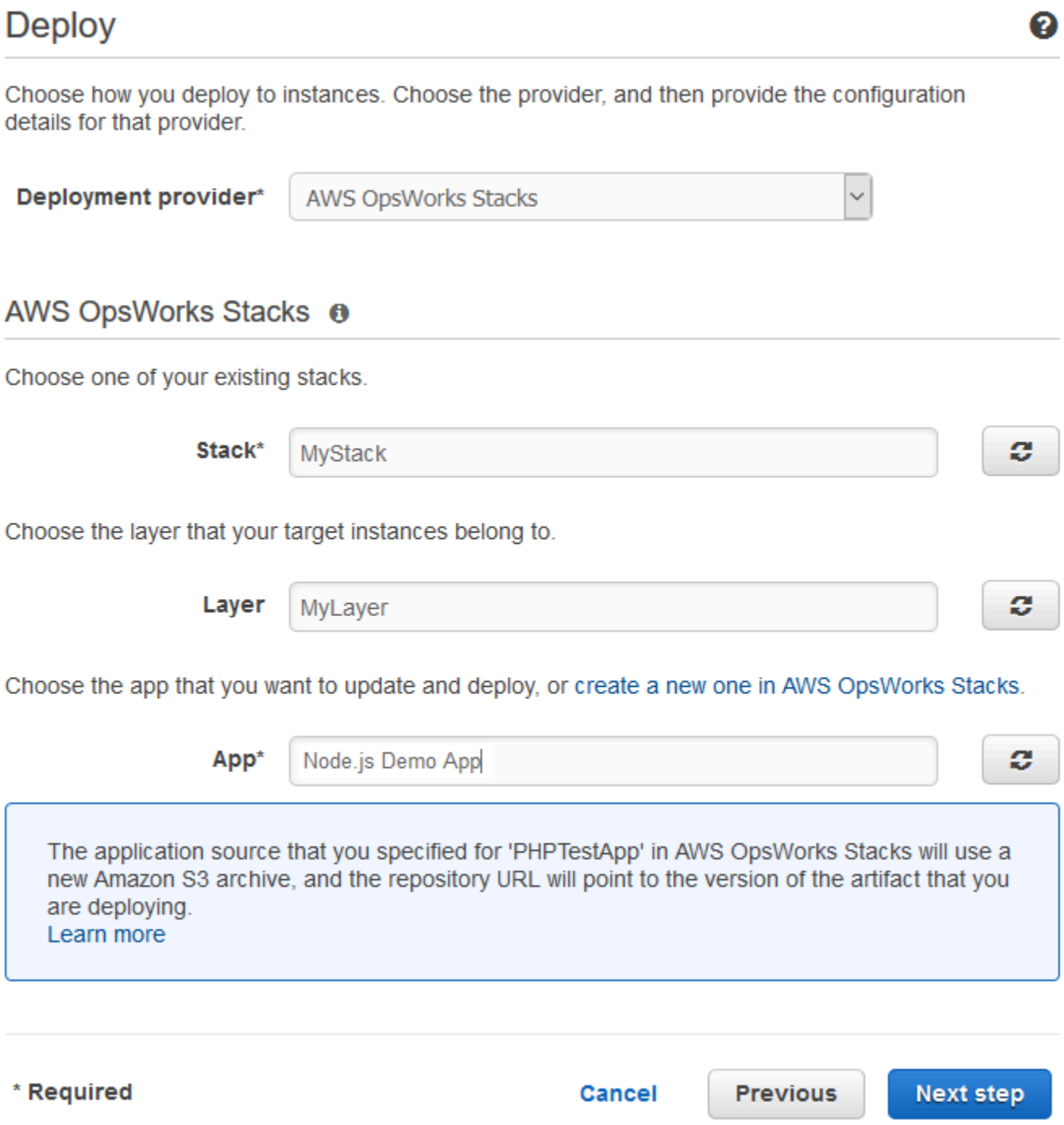
Une fois que vous avez configuré une pile avec une couche et au moins une instance dans AWS OpsWorks Stacks, créez un pipeline CodePipeline avec AWS OpsWorks Stacks comme fournisseur pour déployer des applications ou des livres de recettes Chef sur vos AWS OpsWorks ressources Stacks.

Pour créer un pipeline

1. Ouvrez la CodePipeline console à l'adresse https://console.aws.amazon.com/codepipeline/.
2. Choisissez Créer un pipeline.
3. Sur la page Mise en route avec CodePipeline, saisissez **`MyOpsWorksPipeline`** ou autre nom de pipeline spécifique à votre compte, puis choisissez Étape suivante.
4. Sur la page Emplacement source, sélectionnez Amazon S3 dans la liste déroulante Fournisseur de source.
5. Dans la zone de détails d'Amazon S3, saisissez le chemin de votre compartiment Amazon S3, au format suivant**`s3://`*`bucket-name`*`/`*`file name`***. Reportez-vous au lien que vous avez noté à l'étape 9 de Étape 3 : Charger le code de l'application dans un compartiment Amazon S3. Dans cette procédure pas à pas, le chemin est `s3://my-appbucket/opsworks-nodejs-demo-app.zip`. Choisissez Next step (Étape suivante).



6. Sur la page Génération, choisissez Aucune génération dans la liste déroulante, puis choisissez Étape suivante.
7. Sur la page Deploy (Déployer), choisissez AWS OpsWorks Stacks comme fournisseur de déploiement.



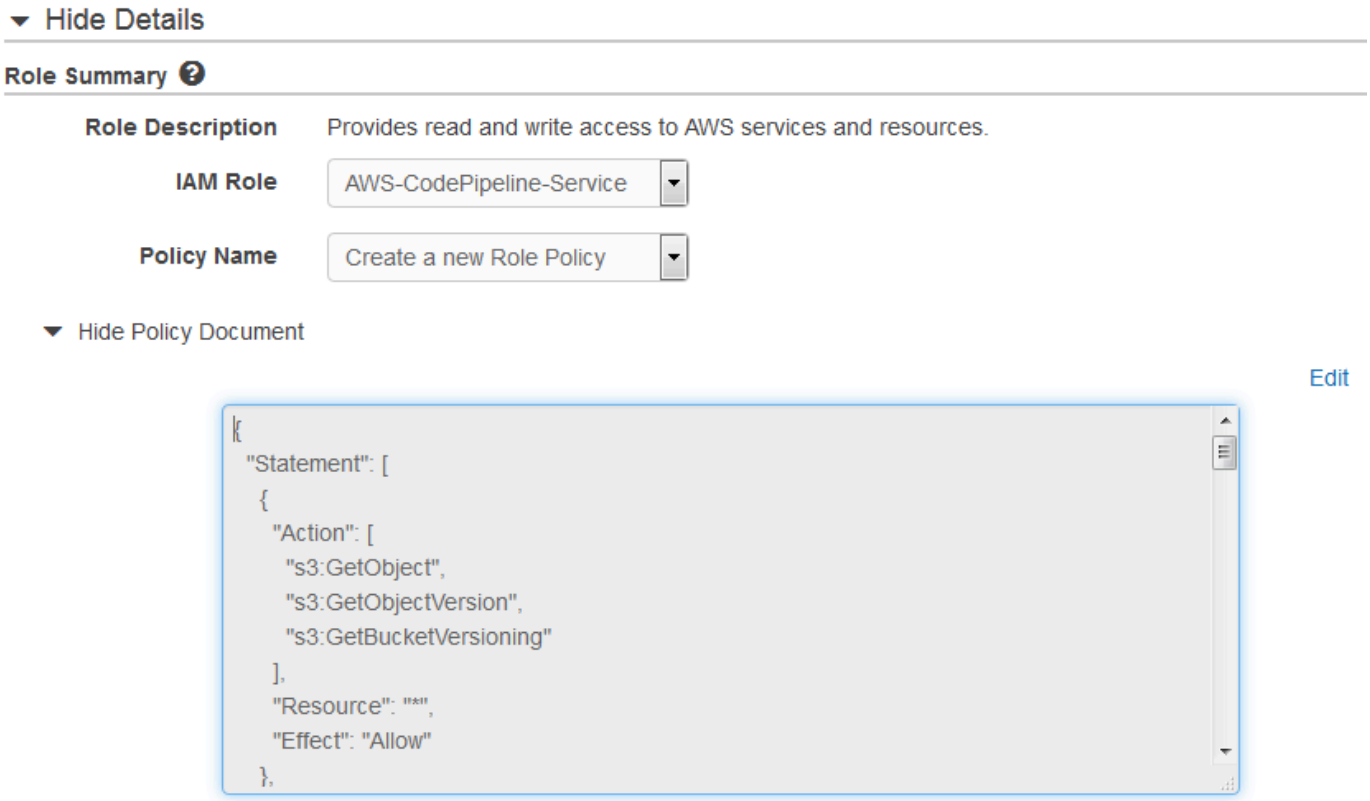
8. Dans le champ Stack (Pile), saisissez `CodePipelineDemo` ou le nom de la pile que vous avez créée dans Étape 1 : créer une pile, une couche et une instance dans AWS OpsWorks Stacks.
9. Dans le champ Couche, saisissez `Node.js App Server` ou le nom de la couche que vous avez créée dans Étape 1 : créer une pile, une couche et une instance dans AWS OpsWorks Stacks.

10. Dans le champ Application, sélectionnez l'application que vous avez téléchargée sur Amazon S3[Étape 3 : Charger le code de l'application dans un compartiment Amazon S3](), puis choisissez Étape suivante.
11. Sur la page Rôle de AWS service, choisissez Créer un rôle.

    Une nouvelle fenêtre s'ouvre avec une page de console IAM qui décrit le rôle qui sera créé pour vous. `AWS-CodePipeline-Service` Dans la liste déroulante Nom de la stratégie, choisissez Créer une stratégie. Veillez à ce que le document de stratégie ait le contenu suivant. Choisissez Modifier pour modifier le document de stratégie, si nécessaire.

```
{
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:GetBucketVersioning"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": "opsworks:*",
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

    Lorsque vous avez terminé d'apporter des modifications au document, cliquez sur Autoriser. Vos modifications seront affichées dans la console IAM.



> **Note**
>
> Si la création d'un rôle échoue, c'est peut-être parce que vous possédez déjà un rôle IAM nommé AWS- CodePipeline -Service. Si vous utilisiez le rôle AWS- CodePipeline -Service avant mai 2016, le rôle n'est peut-être pas autorisé à utiliser AWS OpsWorks Stacks en tant que fournisseur de déploiement. Dans ce cas, vous devez mettre à jour la déclaration de stratégie comme illustré dans cette étape. Si vous voyez un message d'erreur, revenez au début de cette étape et choisissez Utiliser le rôle existant au lieu de Créer un rôle. Si vous utilisez un rôle existant, le rôle doit avoir une stratégie attachée qui inclut les autorisations affichées dans cette étape. Pour plus d'informations sur le rôle de service et sa déclaration de stratégie, consultez Modifier la stratégie d'un rôle de service IAM.

12. Si le processus de création du rôle aboutit, la page IAM se fermera et vous serez renvoyé à la page AWS Service Role. Choisissez Next step (Étape suivante).
13. Sur la page Vérification du pipeline, vérifiez les choix affichés, puis choisissez Créer un pipeline.
14. Lorsque votre pipeline est prêt, il doit commencer à localiser votre code source et à déployer automatiquement votre application sur votre pile. Ce processus peut prendre plusieurs minutes.

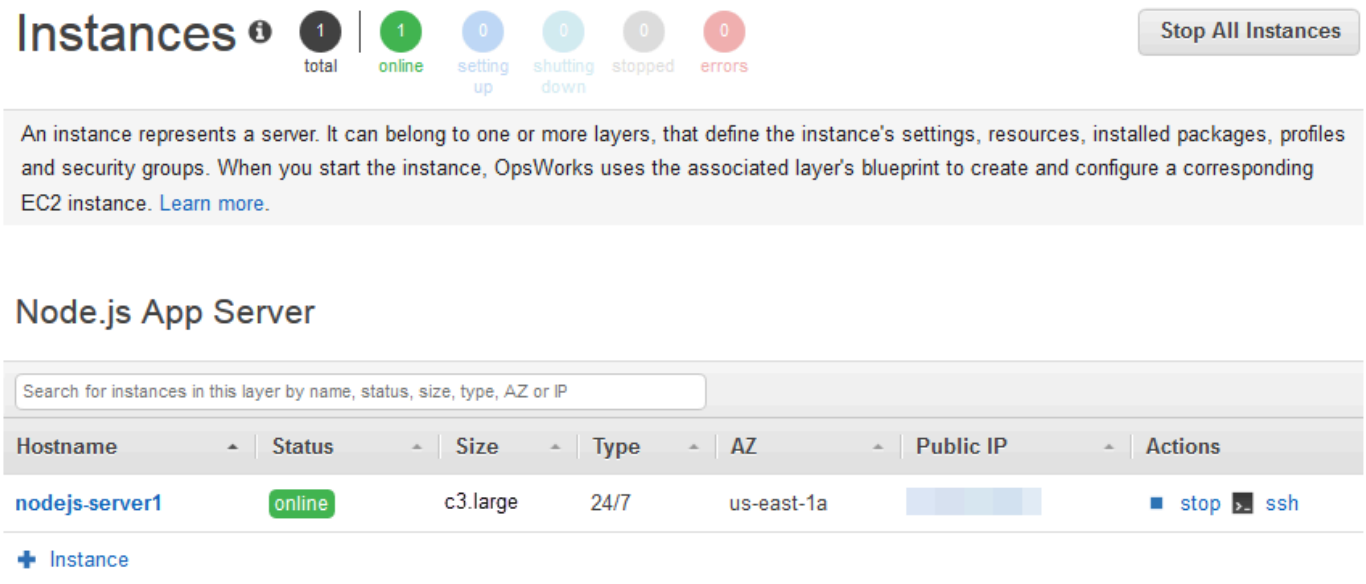
Étape 6 : Vérification du déploiement de l'application dans AWS OpsWorks Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

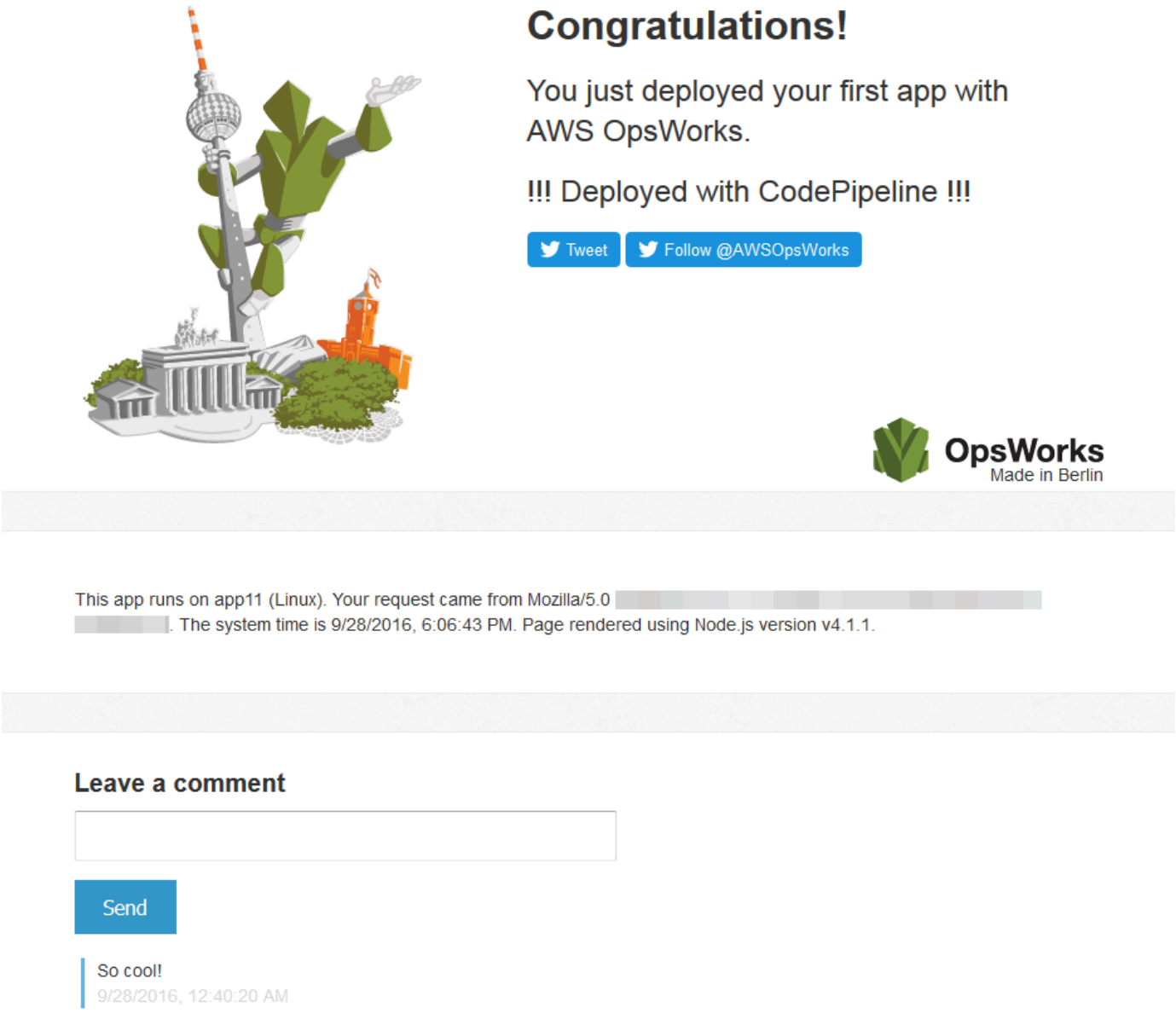
Pour vérifier que l'application Node.js a CodePipeline été déployée sur votre stack, connectez-vous à l'instance dans laquelle vous l'avez crééeÉtape 1 : créer une pile, une couche et une instance dans AWS OpsWorks Stacks. Vous devriez être capable de voir et d'utiliser l'application web Node.js.

Pour vérifier le déploiement de l'application dans votre AWS OpsWorks instance Stacks

1. Ouvrez la AWS OpsWorks console à l'adresse https://console.aws.amazon.com/opsworks/.
2. Sur le tableau de bord AWS OpsWorks Stacks, choisissez CodePipelineDemo, puis choisissez Node.js App Server.
3. Dans le volet de navigation, sélectionnez Instances, puis sélectionnez l'adresse IP publique de l'instance que vous avez créée pour voir l'application web.



L'application apparaît dans un nouvel onglet du navigateur.



Étape 7 (facultatif) : Mettre à jour le code d'application pour voir CodePipeline redéployer votre application automatiquement

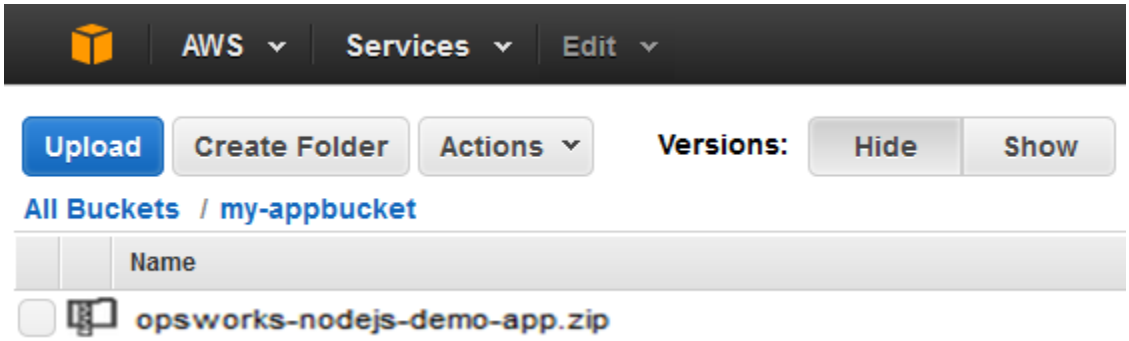
> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Lorsque vous modifiez le code d'applications ou de livres de recettes que vous avez déployés en utilisant CodePipeline, les artefacts mis à jour sont automatiquement déployés sur vos instances cibles (dans ce cas, sur une pile AWS OpsWorks Stacks cible). CodePipeline Cette section vous montre le redéploiement automatique lorsque vous mettez à jour le code dans votre exemple d'application Node.js. Si vous avez toujours le code d'application pour cette procédure pas à pas stockée localement, et que personne d'autre n'a apporté de modifications au code dans la mesure où vous avez entamé la procédure, vous pouvez ignorer les étapes 1 à 4 de cette procédure.

Pour modifier le code de l'exemple d'application

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Ouvrez le compartiment dans lequel vous stockez votre exemple d'application Node.js.

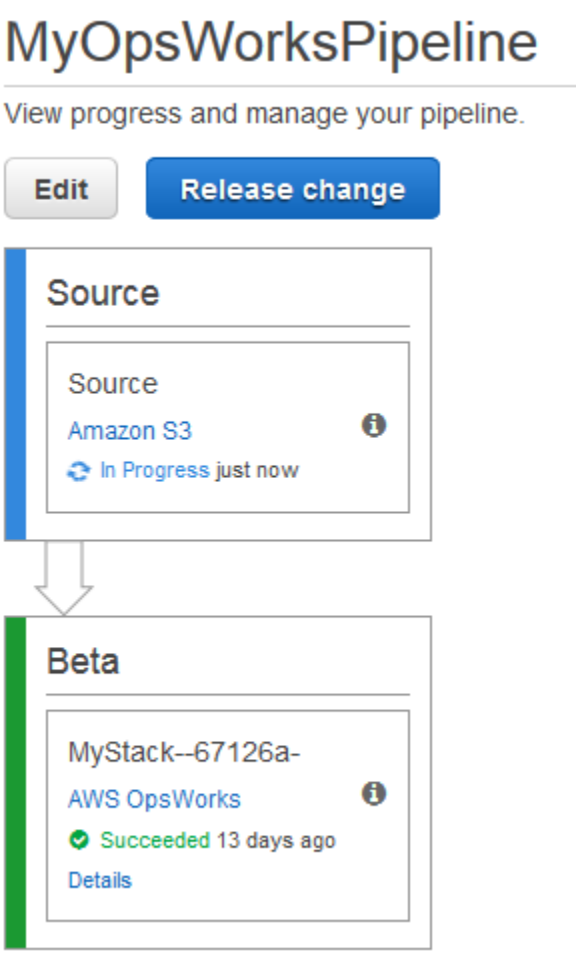


3. Sélectionnez le fichier ZIP qui contient l'application. Dans le menu Actions, sélectionnez Download (Télécharger).
4. Dans la boîte de dialogue, ouvrez le menu contextuel (clic droit), choisissez Télécharger, puis enregistrez le fichier ZIP dans un emplacement approprié. Choisissez OK.
5. Extrayez le contenu du fichier ZIP dans un emplacement approprié. Vous devrez peut-être modifier les autorisations sur le dossier extrait, ainsi que sur les sous-dossiers et sur les contenus, pour autoriser la modification. Dans le dossier `opsworks-nodejs-demo-app \views`, ouvrez le fichier `header.html` pour le modifier.
6. Procédez à une recherche avec les termes `You just deployed your first app with`. Remplacez le mot `deployed` par `updated`. Sur la ligne suivante, remplacez `AWS OpsWorks.` par `AWS OpsWorks and AWS CodePipeline.`Ne modifiez rien d'autre que le texte.

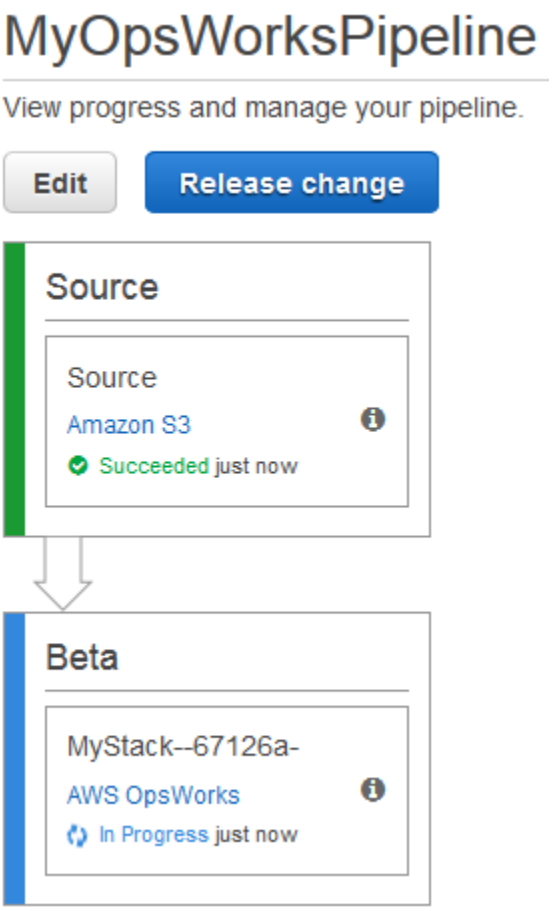
```
<div id="main" role="main">LF
  <div class="container">LF
    <div class="hero-unit">LF
      <div class="robot">LF
        <h1>Congratulations!</h1>LF
        <h2>LF
          You just updated your first app with<br/>LF
          AWS OpsWorks and AWS CodePipeline.LF
        </h2>LF
```

7. Enregistrez et fermez le fichier `header.html`.
8. Compressez le dossier `opsworks-nodejs-demo-app` et enregistrez le fichier ZIP dans un emplacement approprié. Ne modifiez pas le nom du fichier ZIP.
9. Téléchargez le nouveau fichier ZIP dans votre compartiment Amazon S3. Dans cette procédure pas à pas, le nom du compartiment est `my-appbucket`.
10. Ouvrez la CodePipeline console et ouvrez votre pipeline AWS OpsWorks Stacks (MyOpsWorksPipeline). Choisissez Release Change (Modification de version).

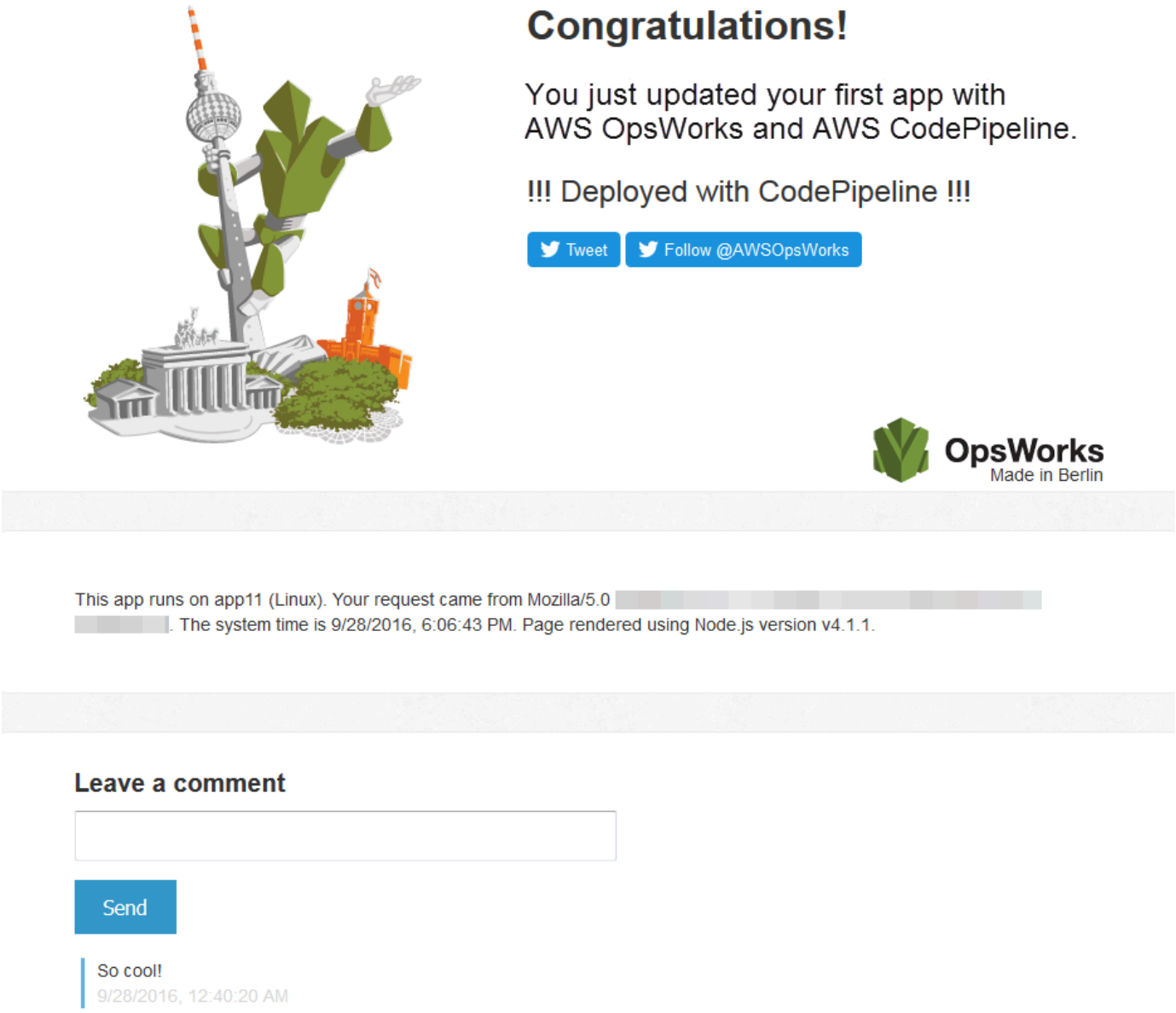
    (Vous pouvez attendre de CodePipeline détecter le changement de code par rapport à la version mise à jour de l'application dans votre compartiment Amazon S3. Pour vous faire gagner du temps, cette procédure pas à pas vous indique simplement de sélectionner Release Change.)
11. Observez les CodePipeline étapes du pipeline au fur et à mesure. CodePipeline Détecte d'abord les modifications apportées à l'artefact source.



CodePipeline envoie le code mis à jour vers votre pile dans AWS OpsWorks Stacks.



12. Lorsque les deux étapes du pipeline se sont terminées avec succès, ouvrez votre pile dans AWS OpsWorks Stacks.
13. Sur la page de propriétés de la pile, choisissez Instances.
14. Dans la colonne IP publique, sélectionnez l'adresse IP publique de votre instance pour afficher le texte de l'application mise à jour.



Étape 8 (facultatif) : Nettoyer les ressources

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Pour éviter que des frais indésirables ne soient débités AWS sur votre compte, vous pouvez supprimer les AWS ressources que vous avez utilisées pour cette procédure pas à pas. Ces AWS ressources incluent la pile AWS OpsWorks Stacks, le rôle IAM et le profil d'instance, ainsi que le pipeline dans lequel vous avez créé. CodePipeline Cependant, vous souhaiterez peut-être continuer à utiliser ces AWS ressources pour en savoir plus sur AWS OpsWorks Stacks et CodePipeline. Si vous souhaitez conserver ces ressources, vous avez terminé cette procédure pas à pas.

Pour supprimer l'application de la pile

Comme vous n'avez ni créé ni appliqué l'application dans le cadre de votre AWS CloudFormation modèle, supprimez l'application de test Node.js avant de supprimer le stack in AWS CloudFormation.

1. Dans la console AWS OpsWorks Stacks, dans le volet de navigation des services, choisissez Applications.
2. Sur la page Applications, sélectionnez Node.js Demo App (Application de démonstration Node.js) et dans Actions, choisissez delete (supprimer). Lorsque vous êtes invité à confirmer la suppression, choisissez Delete (Supprimer). AWS OpsWorks Stacks supprimera l'application.

Pour supprimer la pile

Comme vous avez créé la pile en exécutant un AWS CloudFormation modèle, vous pouvez supprimer la pile, y compris la couche, l'instance, le profil d'instance et le groupe de sécurité créés par le modèle, dans la AWS CloudFormation console.

1. Ouvrez la AWS CloudFormation console.
2. Dans le tableau de bord de la AWS CloudFormation console, sélectionnez la pile que vous avez créée. Dans le menu Actions , sélectionnez Delete Stack (Supprimer la pile). Lorsque vous êtes invité à confirmer, choisissez Oui, supprimer.
3. Attendez que DELETE_COMPLETE apparaisse dans la colonne Statut de la pile.

Pour supprimer le pipeline

1. Ouvrez la CodePipeline console.

2. Dans le CodePipeline tableau de bord, choisissez le pipeline que vous avez créé pour cette procédure.
3. Sur la page du pipeline, choisissez Modifier.
4. Sur la page Edit, choisissez Delete. Lorsque vous êtes invité à confirmer, choisissez Delete.

AWS CodePipeline avec AWS OpsWorks Stacks - Chef 11 Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS CodePipelinevous permet de créer des pipelines de livraison continus qui suivent les modifications de code provenant de sources telles qu' CodeCommitAmazon Simple Storage Service (Amazon S3) ou. GitHub L'exemple présenté dans cette rubrique décrit comment créer et utiliser un pipeline simple CodePipeline comme outil de déploiement pour le code que vous exécutez sur des couches AWS OpsWorks Stacks. Dans cet exemple, vous créez un pipeline pour une application PHP simple, puis vous demandez à AWS OpsWorks Stacks d'exécuter l'application sur toutes les instances d'une couche d'une pile Chef 11.10 (dans ce cas, une seule instance).

> **Note**
>
> Cette rubrique indique comment utiliser un pipeline pour exécuter et mettre à jour une application sur une pile Chef 11.10. Pour plus d'informations sur l'utilisation d'un pipeline pour exécuter et mettre à jour une application sur une pile Chef 12, consultezAWS CodePipeline avec AWS OpsWorks Stacks - Chef 12 Stacks. Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Rubriques

- Prérequis
- Autres scénarios pris en charge
- Étape 1 : Création d'une pile, d'une couche et d'une instance dans AWS OpsWorks Stacks
- Étape 2 : Charger le code de l'application dans un compartiment Amazon S3
- Étape 3 : Ajouter votre AWS OpsWorks application
- Étape 4 : Créer un pipeline CodePipeline
- Étape 5 : Vérification du déploiement de l'application dans AWS OpsWorks Stacks
- Étape 6 (facultatif) : Mettre à jour le code d'application pour voir CodePipeline redéployer votre application automatiquement
- Étape 7 (facultatif) : Nettoyer les ressources

Prérequis

Avant de commencer cette procédure, veillez à disposer des autorisations d'administrateur pour exécuter les tâches suivantes. Vous pouvez être membre d'un groupe auquel la AdministratorAccesspolitique est appliquée, ou vous pouvez être membre d'un groupe doté des autorisations et des politiques indiquées dans le tableau suivant. Pour des raisons de sécurité, vous devez appartenir à un groupe autorisé à effectuer les tâches suivantes, au lieu d'attribuer les autorisations requises à des utilisateurs individuels.

Pour plus d'informations sur la création d'un groupe de sécurité dans IAM et l'attribution d'autorisations au groupe, consultez la section Création de groupes d'utilisateurs IAM. Pour plus d'informations sur la gestion des autorisations AWS OpsWorks Stacks, consultez Bonnes pratiques : gestion des autorisations.

Autorisations	Stratégie recommandée pour attacher à un groupe
Créez et modifiez des piles, des couches et des instances dans AWS OpsWorks Stacks.	AWSOpsWorks_FullAccess
Créez, modifiez et exécutez les modèles dans AWS CloudFormation.	AmazonCloudFormationFullAccess
Créez, modifiez et accédez aux compartiments Amazon S3.	Amazon S3 FullAccess

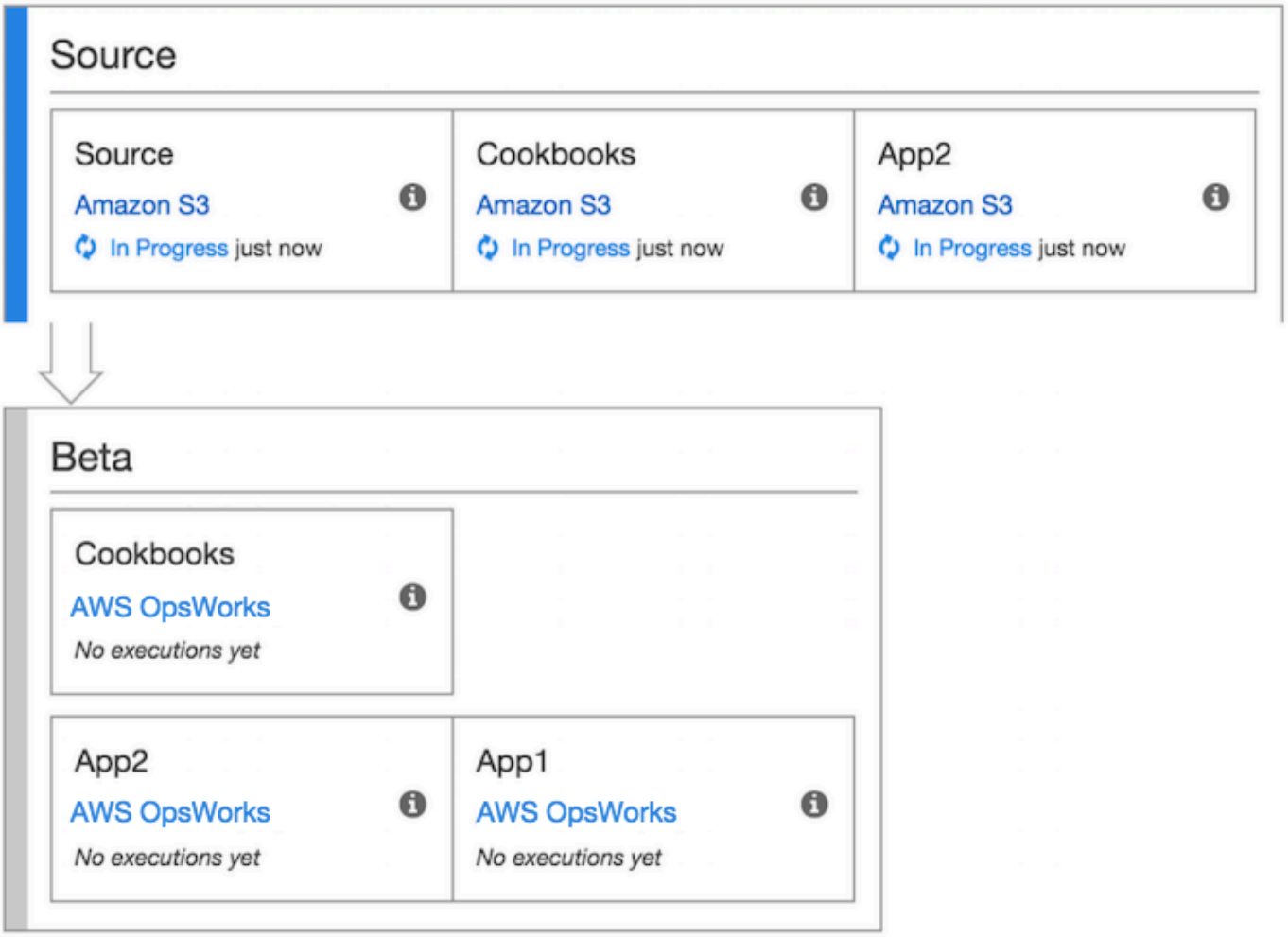
Autorisations	Stratégie recommandée pour attacher à un groupe
Créez, modifiez et exécutez des pipelines CodePipeline, en particulier des pipelines qui utilisent AWS OpsWorks Stacks comme fournisseur.	AWSCodePipeline_FullAccess

Vous devez également disposer d'une paire de EC2 clés Amazon. Dans cette procédure pas à pas, vous serez invité à fournir le nom de cette paire de clés lorsque vous exécuterez le AWS CloudFormation modèle qui crée la pile d'échantillons, la couche et l'instance. Pour plus d'informations sur l'obtention d'une paire de clés dans la EC2 console Amazon, consultez la section Créer une paire de clés dans la EC2 documentation Amazon. La paire de clés doit résider dans la région USA Est (Virginie du Nord). Vous pouvez utiliser une paire de clés existante, si vous en avez déjà une dans cette région.

Autres scénarios pris en charge

Cette procédure crée un simple pipeline qui inclut une étape Source et une étape Deploy (Déployer). Cependant, vous pouvez créer des pipelines plus complexes qui utilisent AWS OpsWorks Stacks comme fournisseur. Les exemples suivants concernent les pipelines et les scénarios pris en charge :

- Vous pouvez modifier un pipeline pour ajouter un livre de recettes Chef à l'étape Source et une cible associée pour les livres de recettes mis à jour à l'étape Deploy (Déployer). Dans ce cas, vous ajoutez une action Deploy (Déployer) qui déclenche la mise à jour de vos livres de recettes lorsque vous apportez des modifications à la source. Le livre de recettes mis à jour est déployé avant votre application.
- Vous pouvez créer un pipeline complexe, avec des livres de recettes personnalisés et plusieurs applications, et le déployer sur une AWS OpsWorks pile Stacks. Le pipeline suit les modifications apportées à la fois aux sources de l'application et aux sources des livres de recettes, et redéploie lorsque vous avez apporté les modifications. Vous trouverez ci-dessous un exemple de pipeline complexe et similaire:



Pour plus d'informations sur l'utilisation CodePipeline, consultez la CodePipelinedocumentation.

Étape 1 : Création d'une pile, d'une couche et d'une instance dans AWS OpsWorks Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Pour utiliser AWS OpsWorks Stacks comme fournisseur de déploiement pour un pipeline, vous devez d'abord disposer d'une pile, d'une couche et d'au moins une instance dans la couche. Bien que vous puissiez créer une pile dans AWS OpsWorks Stacks en suivant les instructions de Getting Started with Linux Stacks ou Getting Started with Windows Stacks, pour gagner du temps, cet exemple utilise un AWS CloudFormation modèle pour créer une pile, une couche et une instance Chef 11.10 basées sur Linux. L'instance créée par ce modèle exécute Amazon Linux 2016.03 et possède `c3.large` comme type d'instance.

> **Important**
>
> Le AWS CloudFormation modèle doit être stocké et exécuté dans la même région que le compartiment Amazon S3 dans lequel vous téléchargerez ultérieurement votre application et dans la même région dans laquelle vous créerez ultérieurement votre pipeline CodePipeline. Pour le moment, CodePipeline prend uniquement en charge le fournisseur AWS OpsWorks Stacks dans la région USA Est (Virginie du Nord) (us-east-1). Toutes les ressources de cette procédure doivent être créées dans la région USA Est (Virginie du Nord).
>
> Si la création de la pile échoue, il se peut que vous approchiez du nombre maximal de rôles IAM pour votre compte. La création de piles peut également échouer si votre compte ne peut pas lancer les instances avec un type d'instance `c3.large`. Par exemple, si vous utilisez le niveau AWS gratuit, vous pouvez recevoir un message d'erreur tel que`Root device type: must be included in EBS`. Si votre compte comporte des limites quant aux types d'instances que vous êtes autorisé à créer, telles que les limites imposées par le niveau AWS gratuit, essayez de remplacer la valeur du `InstanceType` paramètre dans le bloc d'instances du modèle par un type d'instance que votre compte peut utiliser.

Pour créer une pile, une couche et une instance à l'aide de AWS CloudFormation

1. Copiez le AWS CloudFormation modèle suivant dans un nouveau document en texte brut. Enregistrez le fichier à un emplacement approprié sur votre ordinateur local et nommez-le NewOpsWorksStack.template, ou un autre nom qui vous convient.

```
{
  "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"
      ]
```

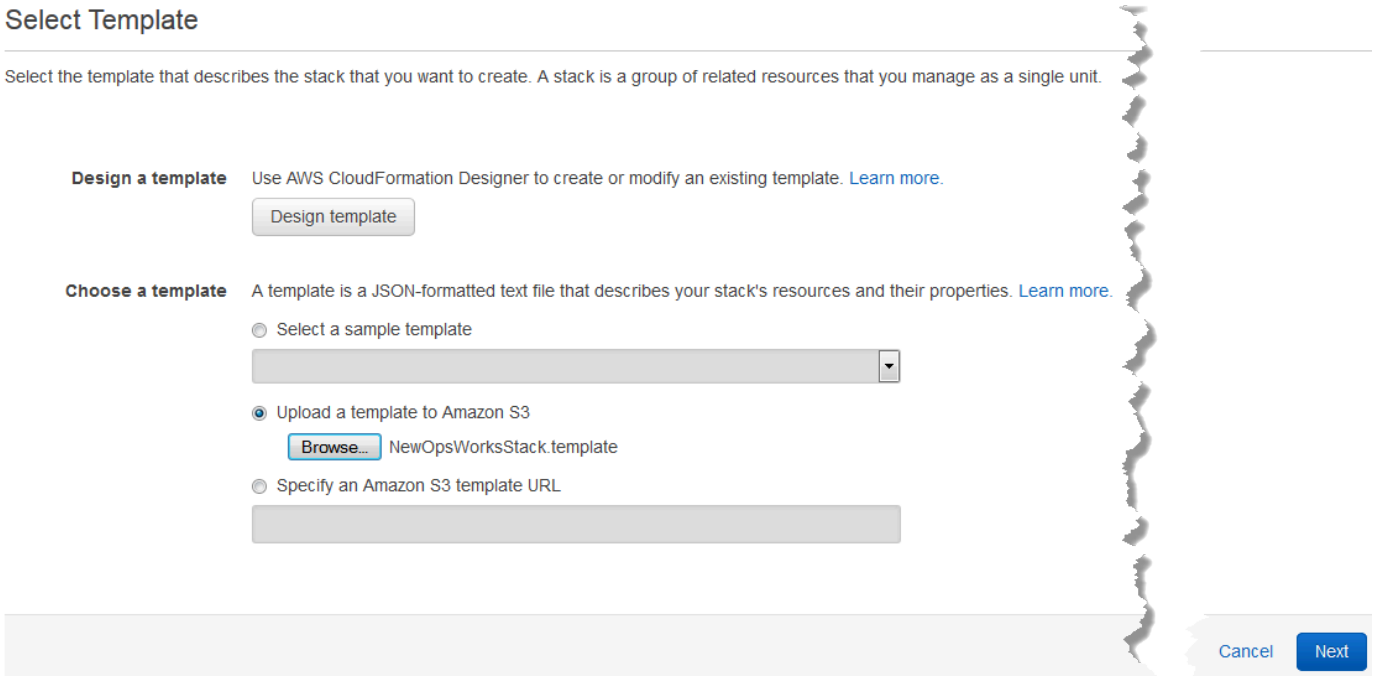
```
    },
    "OpsWorksServiceRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "Service": [
                  {
                    "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. Connectez-vous à la AWS CloudFormation console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudformation.
3. Sur la page d' AWS CloudFormation accueil, choisissez Create stack.
4. Sur la page Sélectionner un modèle, dans la zone Choisir un modèle, choisissez Télécharger un modèle sur Amazon S3, puis Parcourir.
5. Accédez au AWS CloudFormation modèle que vous avez enregistré à l'étape 1, puis choisissez Ouvrir. Sur la page Select Template, choisissez Next.

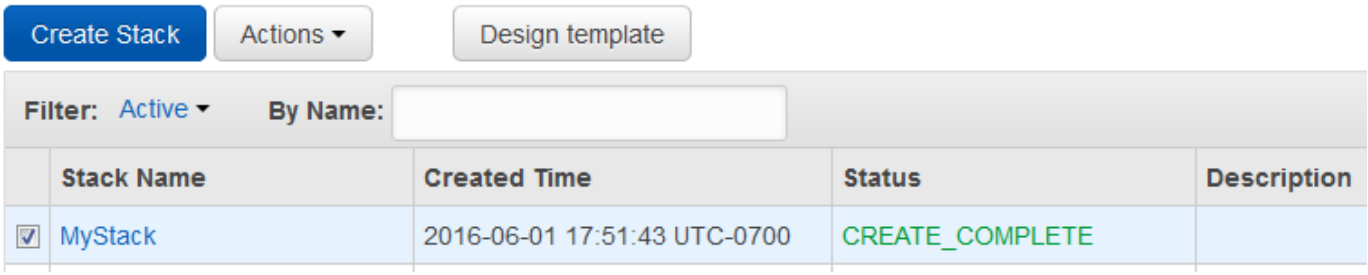


6. Sur la page Spécifier les détails, nommez la pile MyStackou tout autre nom de pile propre à votre compte. Si vous choisissez un autre nom pour votre pile, modifiez le nom de la pile tout au long de cette procédure pas à pas.
7. Dans la zone Paramètres, indiquez le nom d'une paire de EC2 clés que vous souhaitez utiliser pour accéder à votre instance AWS OpsWorks Stacks une fois celle-ci créée. Choisissez Suivant.
8. Dans la page Options, choisissez Suivant. (Les paramètres de cette page ne sont pas obligatoires pour cette procédure pas à pas.)
9. Le AWS CloudFormation modèle que vous utilisez dans cette procédure pas à pas crée des rôles IAM, un profil d'instance et une instance.

> **Important**
>
> Avant de choisir Créer, choisissez Coût pour estimer les frais que vous pourriez encourir AWS pour créer des ressources avec ce modèle.

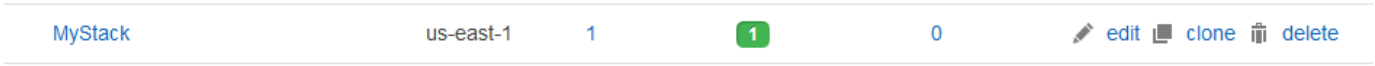
Si la création de ressources IAM est acceptable, cochez la case Je reconnais que ce modèle peut amener AWS CloudFormation à créer des ressources IAM, puis choisissez Create. Si la création de ressources IAM n'est pas acceptable, vous ne pouvez pas poursuivre cette procédure.

10. Le AWS CloudFormation tableau de bord vous permet de visualiser la progression de la création de la pile. Avant de passer à l'étape suivante, patientez jusqu'à ce que CREATE_COMPLETE s'affiche dans la colonne Statut.



Pour vérifier la création d'une pile dans AWS OpsWorks Stacks

1. Ouvrez la AWS OpsWorks console à l'adresse https://console.aws.amazon.com/opsworks/.
2. Sur le tableau de bord AWS OpsWorks Stacks, visualisez la pile que vous avez créée.



3. Ouvrez la pile, puis affichez la couche et l'instance. Notez que la couche et l'instance ont été créées avec les noms et autres métadonnées fournis dans le AWS CloudFormation modèle. Vous êtes prêt à charger votre application dans un compartiment Amazon S3.

Étape 2 : Charger le code de l'application dans un compartiment Amazon S3

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Comme vous devez fournir un lien vers le référentiel de votre code dans le cadre de l'installation du pipeline, préparez le référentiel avant de créer le pipeline. Dans cette procédure, vous chargez une application PHP dans un compartiment Amazon S3.

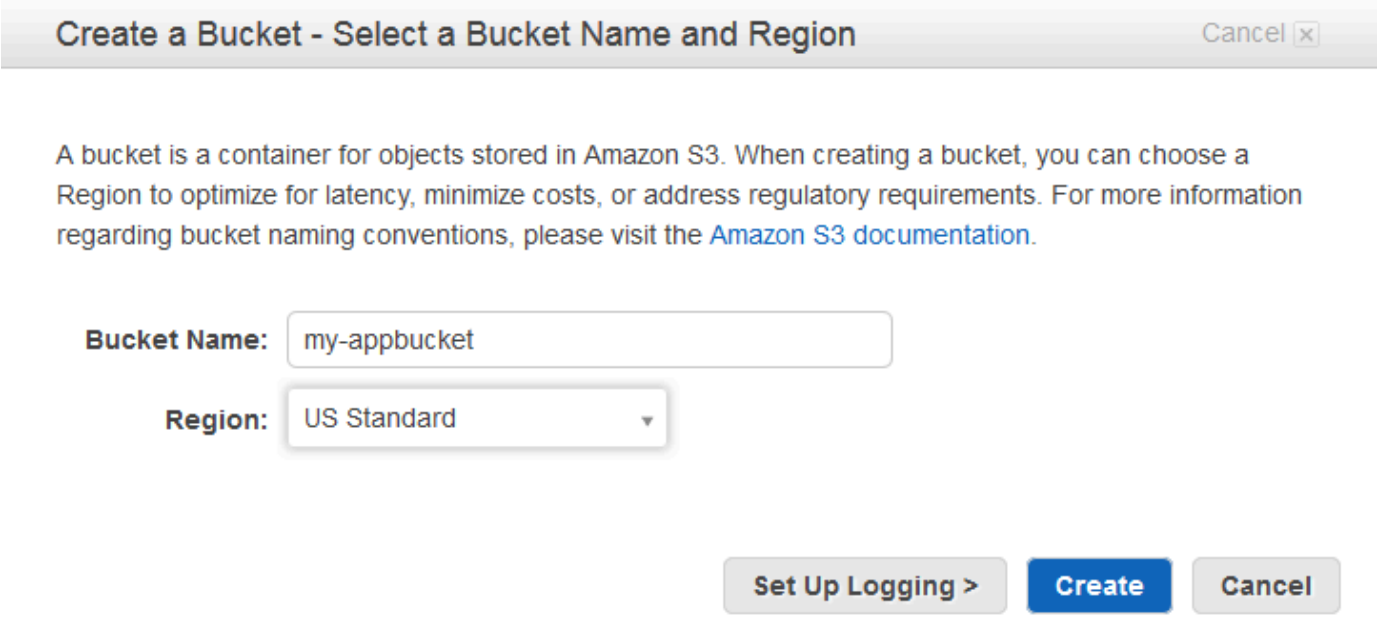
Bien qu'il soit CodePipeline possible d'utiliser du code directement depuis GitHub ou CodeCommit en tant que source, cette procédure pas à pas explique comment utiliser un compartiment Amazon S3. Le compartiment Amazon S3 permet CodePipeline de détecter les modifications apportées au code de l'application et de déployer automatiquement l'application modifiée. Si vous le souhaitez, vous pouvez utiliser un compartiment existant. Assurez-vous que le compartiment répond aux critères CodePipeline décrits dans la section Procédure pas à pas simple du pipeline (compartiment Amazon S3) de la CodePipeline documentation.

> **Important**
>
> Le compartiment Amazon S3 doit se trouver dans la même région que celle dans laquelle vous créerez votre pipeline. Pour le moment, CodePipeline prend uniquement en charge le fournisseur AWS OpsWorks Stacks dans la région USA Est (Virginie du Nord) (us-east-1). Toutes les ressources de cette procédure doivent être créées dans la région USA Est (Virginie du Nord). Le bucket doit également être versionné car il CodePipeline nécessite une source versionnée. Pour plus d'informations, consultez Utilisation de la gestion des versions.

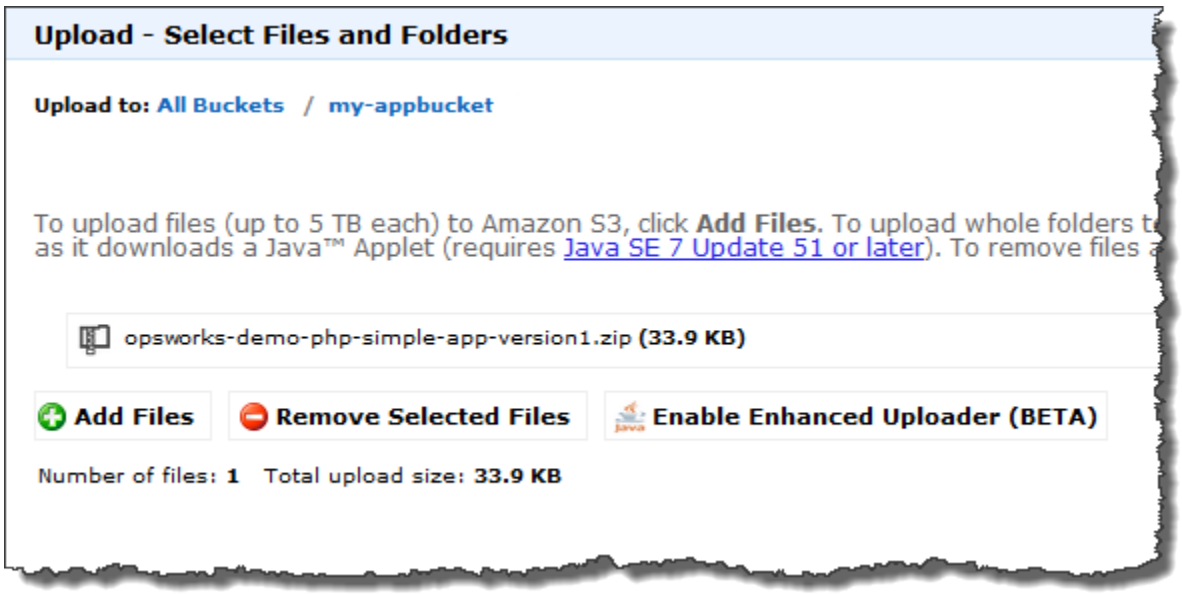
Pour télécharger votre application dans un compartiment Amazon S3

1. GitHub Sur le site Web, téléchargez un fichier ZIP de l'exemple d'application PHP AWS OpsWorks Stacks et enregistrez-le à un emplacement pratique sur votre ordinateur local.

2. Assurez-vous que les dossiers `index.php` et `ASSETS` se trouvent au niveau racine du fichier ZIP téléchargé. S'ils ne le sont pas, décompressez le fichier et créer un nouveau fichier ZIP ayant ces fichiers au niveau racine.
3. Ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
4. Choisissez Créer un compartiment.
5. Sur la page Créer un compartiment - Sélectionner un nom et une région de compartiment, pour Bucket Name (Nom du compartiment), saisissez un nom unique pour votre compartiment. Les noms de compartiment doivent être uniques dans tous les AWS comptes, et pas seulement dans votre propre compte. Cette procédure utilise le nom **`my-appbucket`**, mais vous pouvez utiliser `my-appbucket-`*`yearmonthday`* pour que votre nom de compartiment soit unique. Dans la liste déroulante Region, choisissez US Standard, puis Créer. USA Standard est l'équivalent de `us-east-1`.



6. Choisissez le compartiment que vous avez créé à partir de la liste Tous les compartiments.
7. Sur la page du compartiment, choisissez Charger.
8. Sur la page Charger - Sélectionner les fichiers et dossiers, choisissez Ajouter des fichiers. Accédez au fichier ZIP enregistré à l'étape 1, choisissez Ouvrir, puis choisissez Commencer le chargement.



9. Une fois le chargement terminé, sélectionnez le fichier ZIP dans la liste des fichiers de votre compartiment, puis choisissez Propriétés.
10. Dans le volet Propriétés, copiez le lien vers votre fichier ZIP et notez le lien. Vous aurez besoin du nom de compartiment et de la partie nom de fichier ZIP du lien pour créer votre pipeline.

Étape 3 : Ajouter votre AWS OpsWorks application

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Avant de créer un pipeline CodePipeline, ajoutez l'application de test PHP à AWS OpsWorks Stacks. Lorsque vous créez le pipeline, vous devez sélectionner l'application que vous avez ajoutée à AWS OpsWorks Stacks.

Préparez le lien vers le compartiment Amazon S3 de l'étape 10 de la procédure précédente. Vous aurez besoin du lien au compartiment dans lequel vous avez stocké votre application de test pour terminer cette procédure.

Pour ajouter une application à AWS OpsWorks Stacks

1. Dans la console AWS OpsWorks Stacks MyStack, ouvrez et dans le volet de navigation, sélectionnez Apps.
2. Choisissez Add app (Ajouter une application).
3. Sur la page Add App (Ajouter une application), indiquez les informations suivantes :

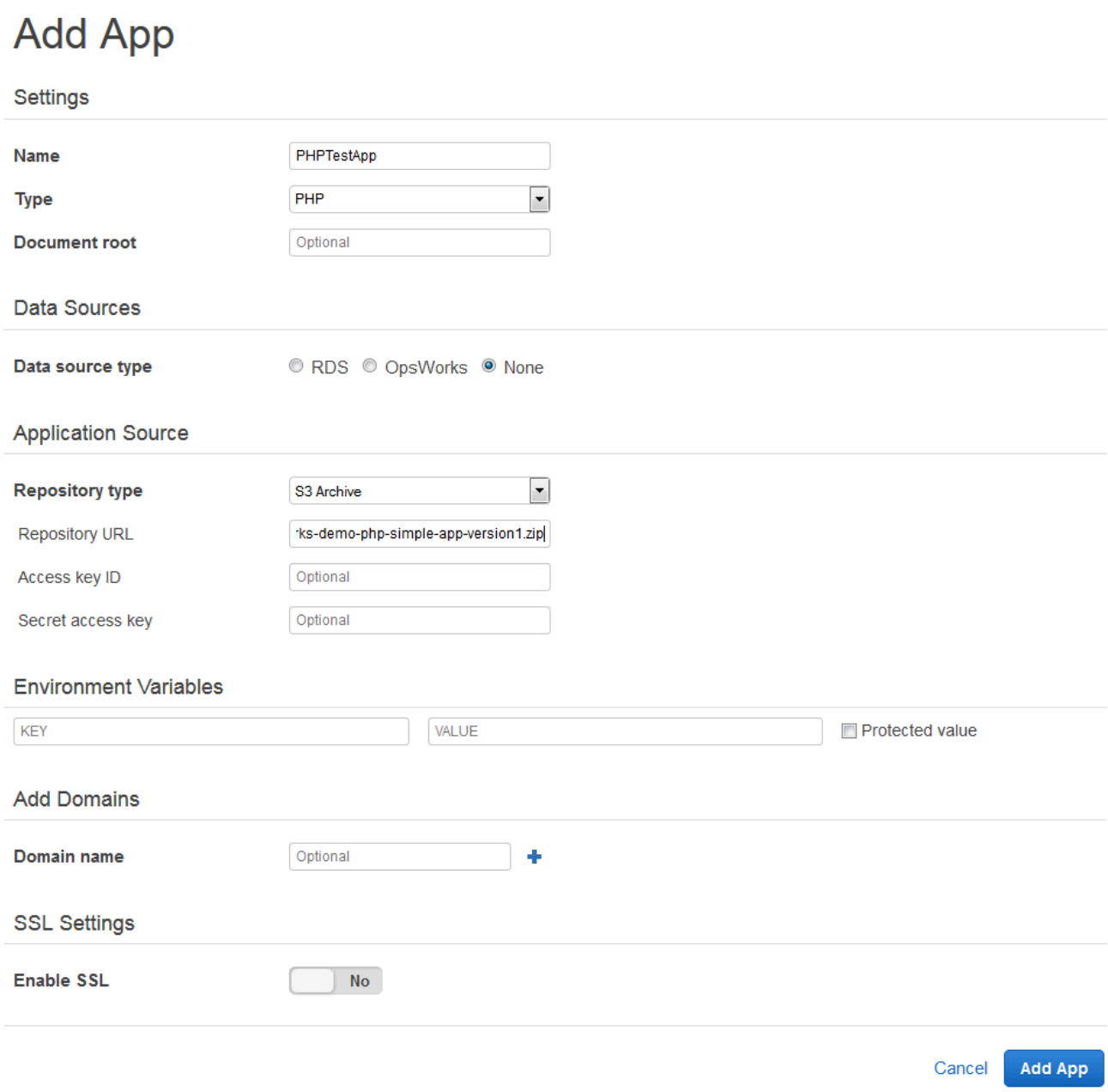
   a. Spécifiez un nom pour votre application. Cette procédure utilise le nom `PHPTestApp`.

   b. Dans la liste déroulante Type, choisissez PHP.

   c. Pour Type de source de données, choisissez Aucun. Cette application ne requiert pas une source de données ou base de données externe.

   d. Dans la liste déroulante Type de référentiel, choisissez Archive S3.

   e. Dans la zone URL du référentiel, collez l'URL que vous avez copiée à l'étape 10 de Étape 2 : Charger le code de l'application dans un compartiment Amazon S3. Votre formulaire doit être similaire à ce qui suit:



4. Vous n'avez pas besoin de modifier d'autres paramètres du formulaire. Choisissez Add App (Ajouter une application).
5. Lorsque l'PHPTestapplication apparaît dans la liste de la page Applications, passez à la procédure suivante,Étape 4 : Créer un pipeline CodePipeline.

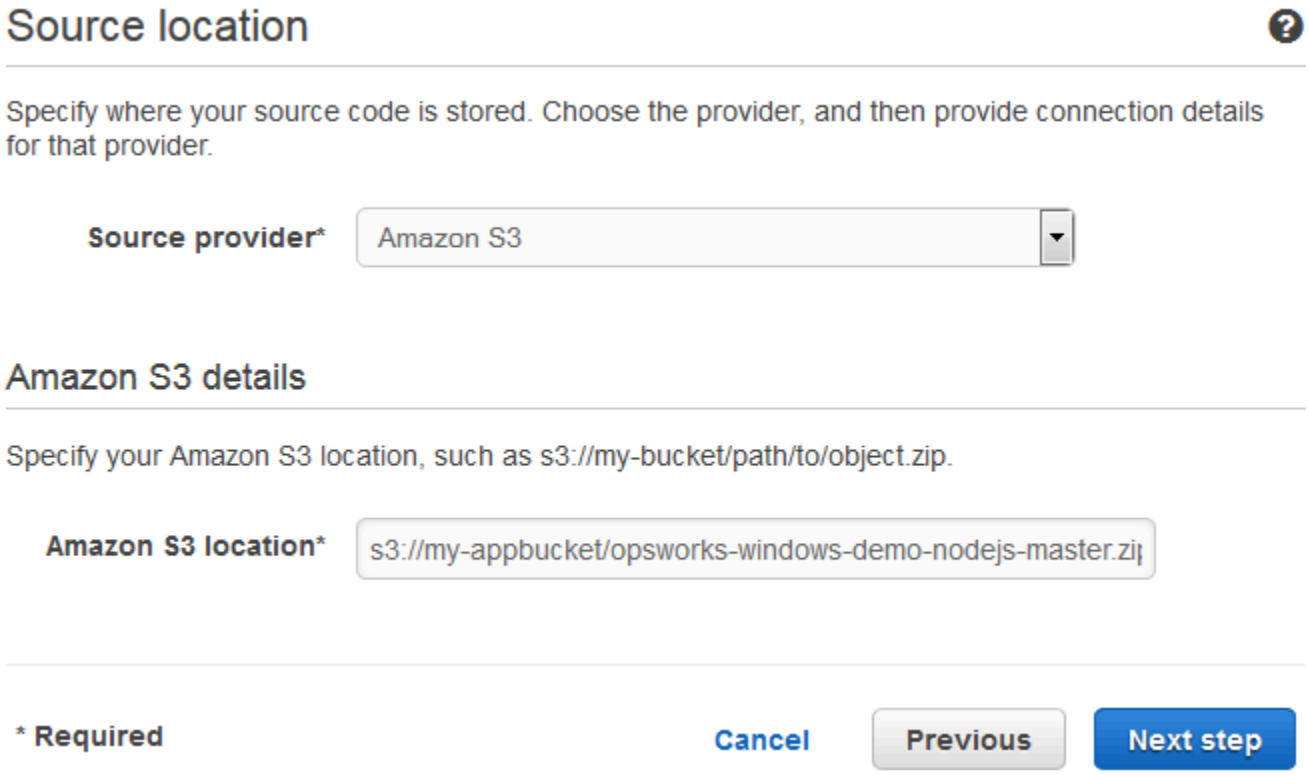
Étape 4 : Créer un pipeline CodePipeline

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

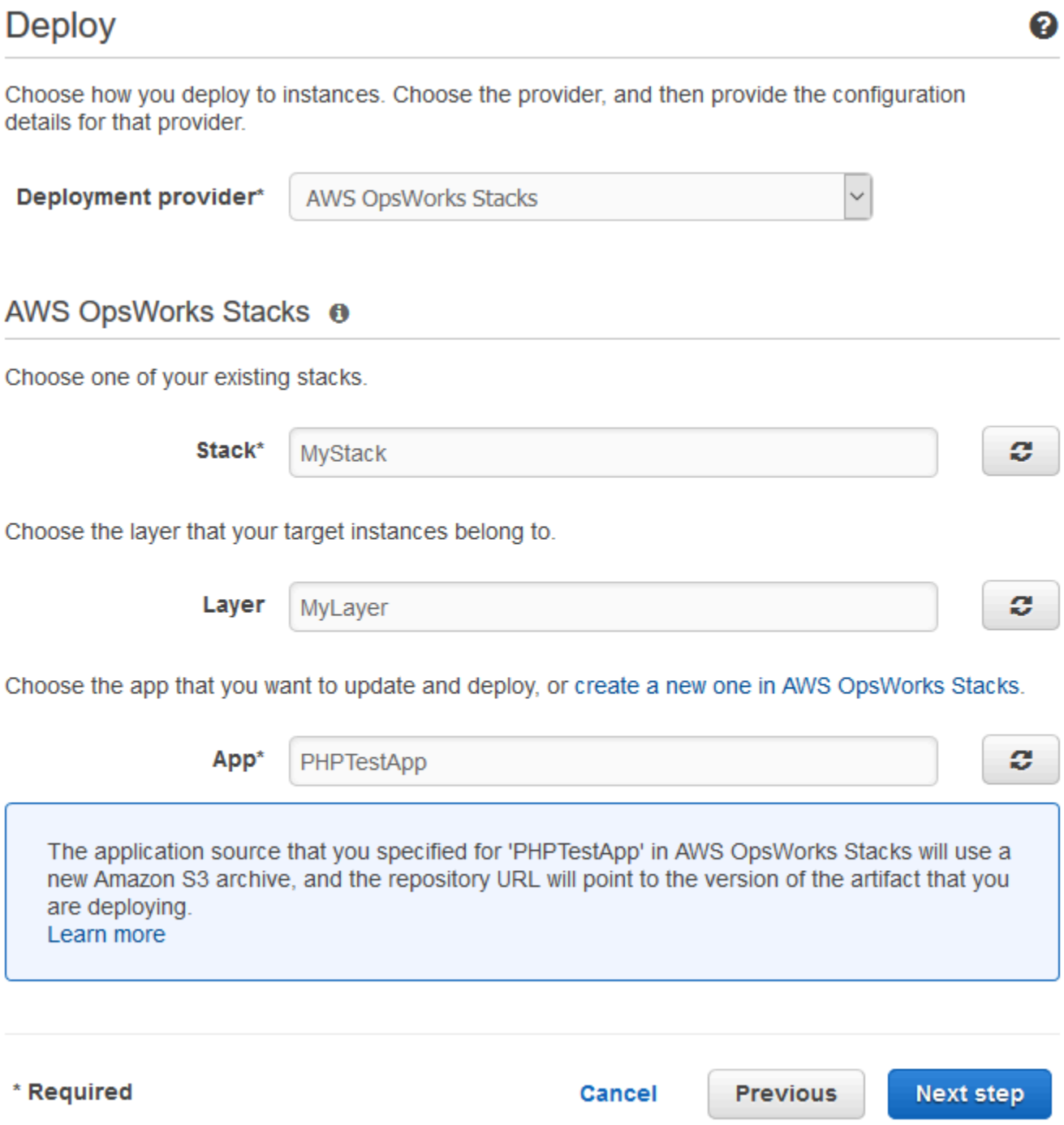
Une fois que vous avez configuré une pile avec une couche et au moins une instance dans AWS OpsWorks Stacks, créez un pipeline CodePipeline avec AWS OpsWorks Stacks comme fournisseur pour déployer des applications ou des livres de recettes Chef sur vos AWS OpsWorks ressources Stacks.

Pour créer un pipeline

1. Ouvrez la CodePipeline console à l'adresse https://console.aws.amazon.com/codepipeline/.
2. Choisissez Créer un pipeline.
3. Sur la page Mise en route avec CodePipeline, saisissez **`MyOpsWorksPipeline`** ou autre nom de pipeline spécifique à votre compte, puis choisissez Étape suivante.
4. Sur la page Emplacement source, sélectionnez Amazon S3 dans la liste déroulante Fournisseur de source.
5. Dans la zone de détails d'Amazon S3, saisissez le chemin de votre compartiment Amazon S3 au format suivant**`s3://`*`bucket-name`*`/`*`file name`***. Reportez-vous au lien que vous avez noté à l'étape 10 de Étape 2 : Charger le code de l'application dans un compartiment Amazon S3. Dans cette procédure pas à pas, le chemin est `s3://my-appbucket/opsworks-demo-php-simple-app-version1.zip`. Choisissez Next step (Étape suivante).



6. Sur la page Génération, choisissez Aucune génération dans la liste déroulante, puis choisissez Étape suivante.
7. Sur la page Deploy (Déployer), choisissez AWS OpsWorks Stacks comme fournisseur de déploiement.



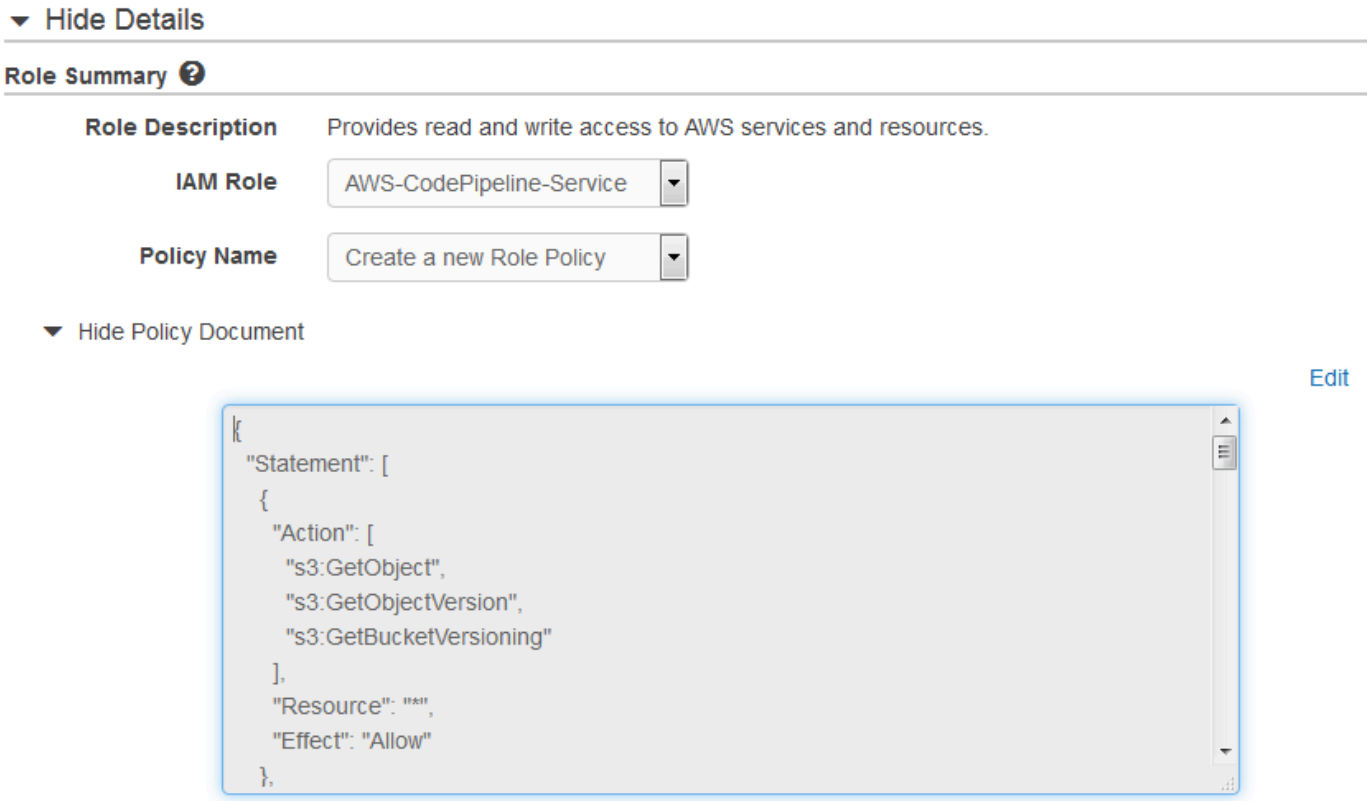
8. Dans le champ Stack (Pile), saisissez `MyStack` ou le nom de la pile que vous avez créée dans Étape 1 : Création d'une pile, d'une couche et d'une instance dans AWS OpsWorks Stacks.
9. Dans le champ Couche, saisissez `MyLayer` ou le nom de la couche que vous avez créée dans Étape 1 : Création d'une pile, d'une couche et d'une instance dans AWS OpsWorks Stacks.

10. Dans le champ Application, sélectionnez l'application que vous avez téléchargée sur Amazon S3Étape 2 : Charger le code de l'application dans un compartiment Amazon S3, puis choisissez Étape suivante.
11. Sur la page Rôle du service AWS, choisissez Créer un rôle.

    Une nouvelle fenêtre s'ouvre avec une page de console IAM qui décrit le rôle qui sera créé pour vous. `AWS-CodePipeline-Service` Dans la liste déroulante Nom de la stratégie, choisissez Créer une stratégie. Veillez à ce que le document de stratégie ait le contenu suivant. Choisissez Modifier pour modifier le document de stratégie, si nécessaire.

```
{
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:GetBucketVersioning"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": "opsworks:*",
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

    Lorsque vous avez terminé d'apporter des modifications au document, cliquez sur Autoriser. Vos modifications seront affichées dans la console IAM.



> **Note**
>
> Si la création d'un rôle échoue, c'est peut-être parce que vous possédez déjà un rôle IAM nommé AWS- CodePipeline -Service. Si vous utilisiez le rôle AWS- CodePipeline -Service avant mai 2016, le rôle n'est peut-être pas autorisé à utiliser AWS OpsWorks Stacks en tant que fournisseur de déploiement ; dans ce cas, vous devez mettre à jour la déclaration de politique comme indiqué dans cette étape. Si vous voyez un message d'erreur, revenez au début de cette étape et choisissez Utiliser le rôle existant au lieu de Créer un rôle. Si vous utilisez un rôle existant, le rôle doit avoir une stratégie attachée qui inclut les autorisations affichées dans cette étape. Pour plus d'informations sur le rôle de service et sa déclaration de stratégie, consultez Modifier la stratégie d'un rôle de service IAM.

12. Si le processus de création du rôle aboutit, la page IAM se fermera et vous serez redirigé vers la page AWS Service Role. Choisissez Next step (Étape suivante).
13. Sur la page Vérification du pipeline, vérifiez les choix affichés, puis choisissez Créer un pipeline.

We will create your pipeline with the following resources.

Source Stage

Source provider Amazon S3

Amazon S3 location s3://my-appbucket0/opsworks-demo-php-simple-app-version1.zip

Build Stage

Build provider No Build

Beta Stage

Deployment provider AWS OpsWorks

Stack MyStack

App PHPTestApp

Layer MyLayer

Pipeline settings

Pipeline name MyOpsWorksPipeline

Artifact location s3://codepipeline-us-east-
AWS CodePipeline will use this existing S3 bucket to store artifacts for this pipeline. Depending on the size of your artifacts, you might be charged for storage costs. For more information, see Amazon S3 storage pricing.

Role name AWS-CodePipeline-Service

To save this configuration with these resources, choose Create pipeline.

Would you like to create this pipeline?

Cancel Previous Create pipeline

14. Lorsque votre pipeline est prêt, il doit commencer à localiser votre code source et à déployer automatiquement votre application sur votre pile. Ce processus peut prendre plusieurs minutes.

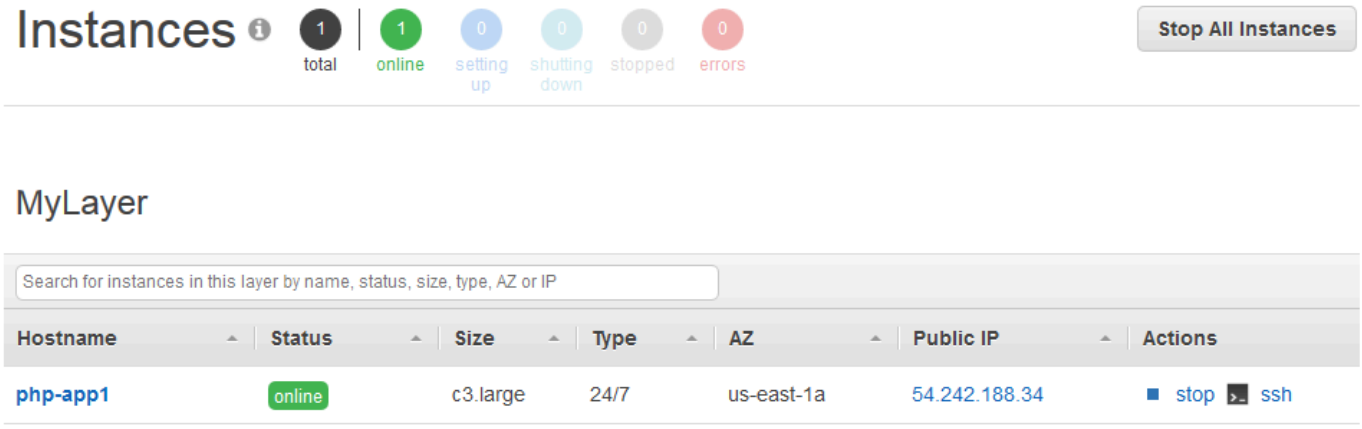
Étape 5 : Vérification du déploiement de l'application dans AWS OpsWorks Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

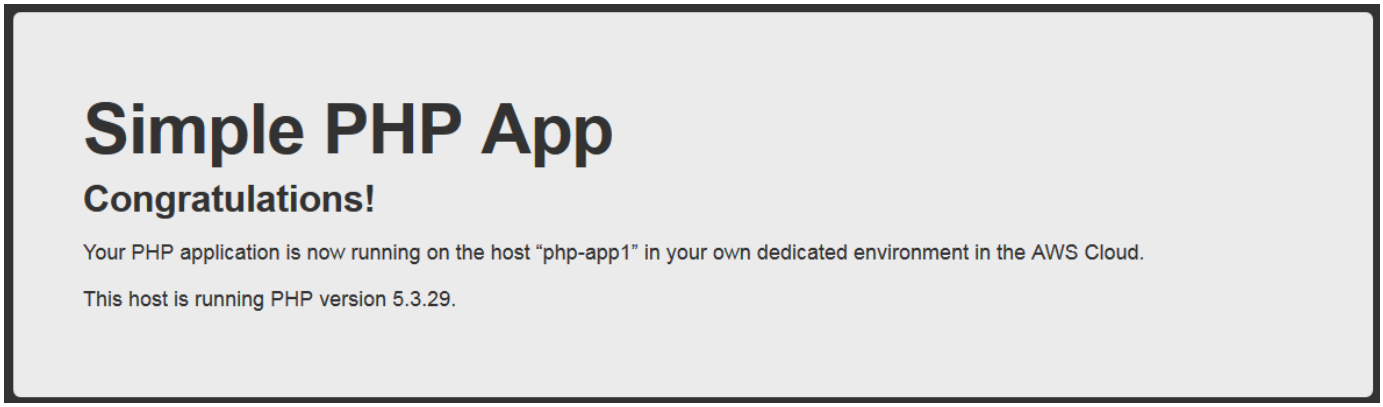
Pour vérifier que l'application PHP a CodePipeline été déployée sur votre stack, connectez-vous à l'instance dans laquelle vous l'avez créée[Étape 1 : Création d'une pile, d'une couche et d'une instance dans AWS OpsWorks Stacks](). Vous devriez être capable de voir et d'utiliser l'application web PHP.

Pour vérifier le déploiement de l'application dans votre AWS OpsWorks instance Stacks

1. Ouvrez la AWS OpsWorks console à l'adresse https://console.aws.amazon.com/opsworks/.
2. Sur le tableau de bord AWS OpsWorks Stacks, choisissez MyStack, puis choisissez MyLayer.
3. Dans le volet de navigation, sélectionnez Instances, puis sélectionnez l'adresse IP publique de l'instance que vous avez créée pour voir l'application web.



L'application apparaît dans un nouvel onglet du navigateur.



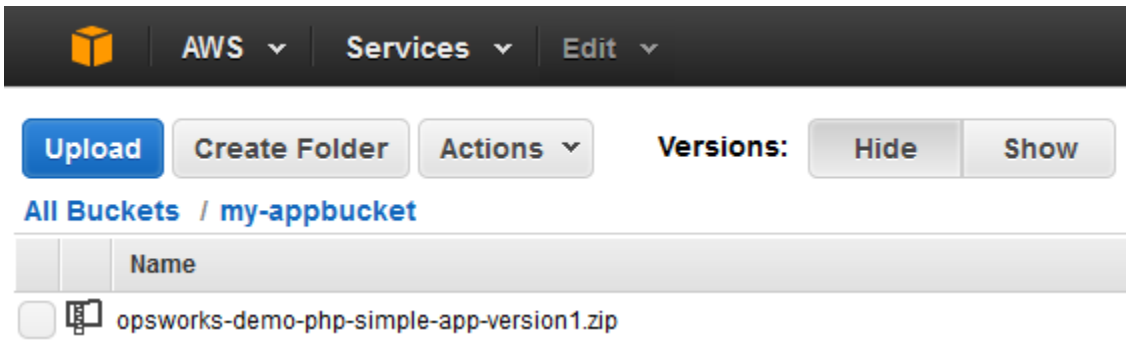
Étape 6 (facultatif) : Mettre à jour le code d'application pour voir CodePipeline redéployer votre application automatiquement

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Lorsque vous modifiez le code d'applications ou de livres de recettes que vous avez déployés en utilisant CodePipeline, les artefacts mis à jour sont automatiquement déployés sur vos instances cibles (dans ce cas, sur une pile AWS OpsWorks Stacks cible). CodePipeline Cette section vous montre le redéploiement automatique lorsque vous mettez à jour le code dans votre exemple d'application PHP.

Pour modifier le code de l'exemple d'application

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Ouvrez le compartiment dans lequel vous stockez votre exemple d'application PHP.



3. Sélectionnez le fichier ZIP qui contient l'application. Dans le menu Actions, sélectionnez Download (Télécharger).
4. Dans la boîte de dialogue, ouvrez le menu contextuel (clic droit), choisissez Télécharger, puis enregistrez le fichier ZIP dans un emplacement approprié. Choisissez OK.
5. Extrayez le contenu du fichier ZIP dans un emplacement approprié. Vous devrez peut-être modifier les autorisations sur le dossier extrait, ainsi que sur les sous-dossiers et sur les contenus, pour autoriser la modification. Dans le dossier `opsworks-demo-php-simple-app-version1`, ouvrez le fichier `index.php` pour le modifier.
6. Procédez à une recherche avec les termes `Your PHP application is now running`. Remplacez le texte `Your PHP application is now running` par `You've just deployed your first app to AWS OpsWorks with AWS CodePipeline,`. Ne modifiez pas les variables.

```
<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>

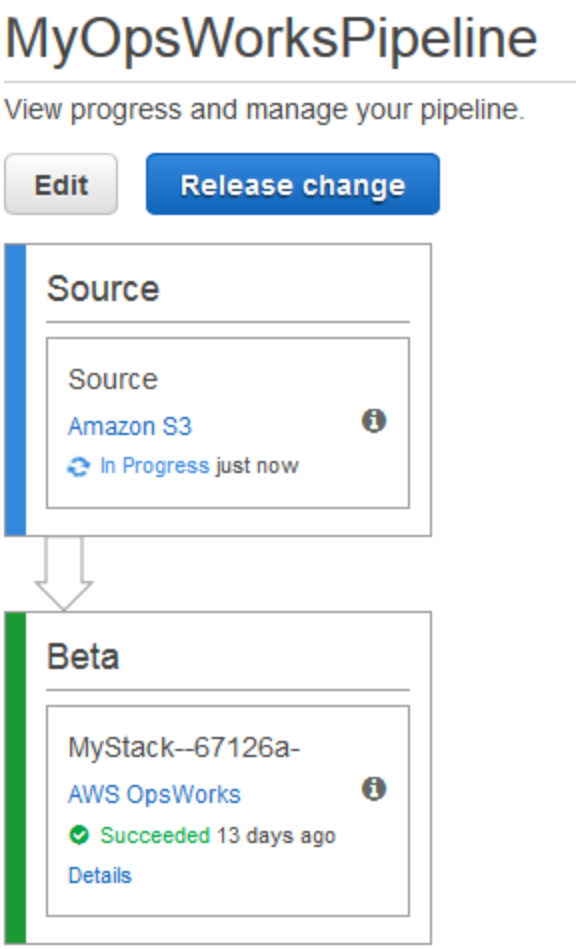
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
    <script src="assets/js/bootstrap.min.js"></script>
</body>
```

7. Enregistrez et fermez le fichier `index.php`.
8. Compressez le dossier `opsworks-demo-php-simple-app-version1` et enregistrez le fichier ZIP dans un emplacement approprié. Ne modifiez pas le nom du fichier ZIP.
9. Téléchargez le nouveau fichier ZIP dans votre compartiment Amazon S3. Dans cette procédure pas à pas, le nom du compartiment est `my-appbucket`.

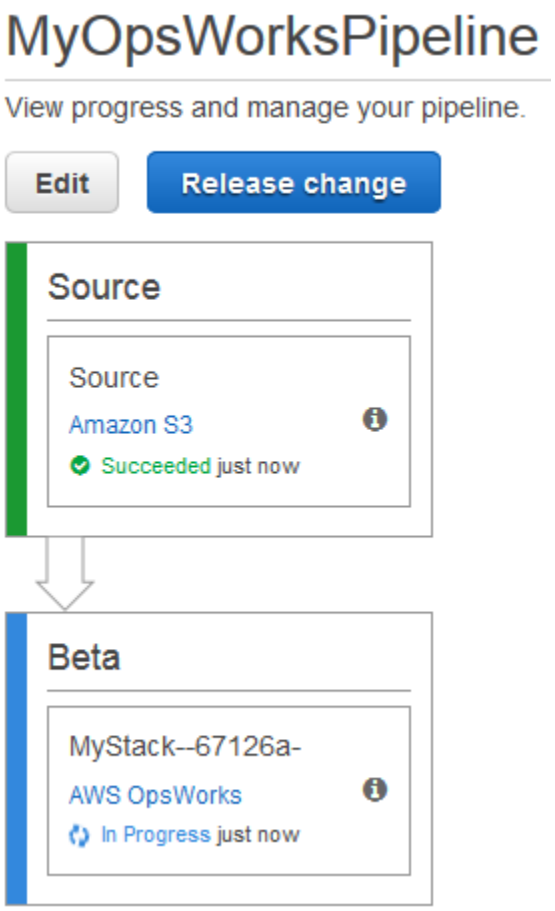
10. Ouvrez la CodePipeline console et ouvrez votre pipeline AWS OpsWorks Stacks (MyOpsWorksPipeline). Choisissez Release Change (Modification de version).

    (Vous pouvez attendre de CodePipeline détecter le changement de code par rapport à la version mise à jour de l'application dans votre compartiment Amazon S3. Pour vous faire gagner du temps, cette procédure pas à pas vous indique simplement de sélectionner Release Change.)

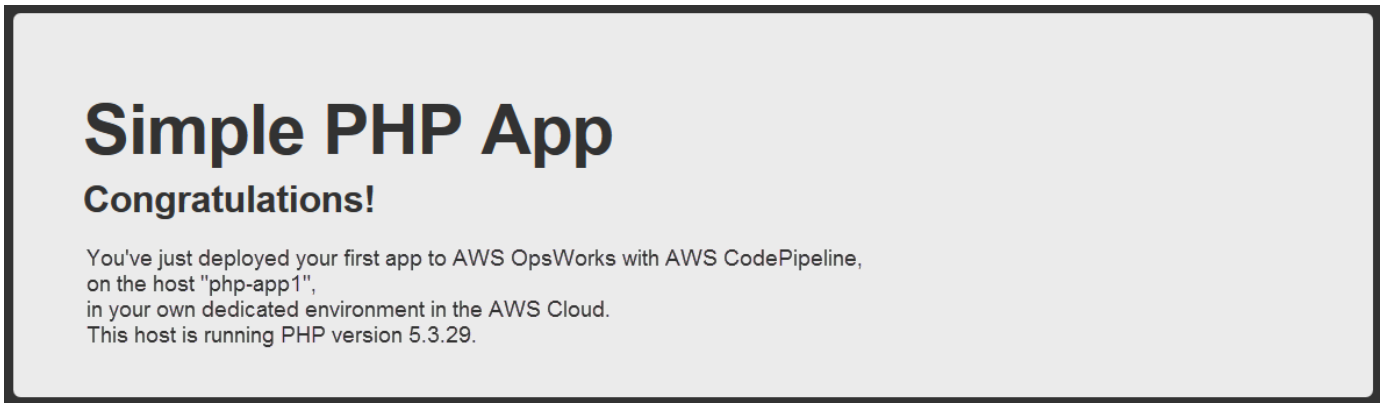
11. Observez les CodePipeline étapes du pipeline au fur et à mesure. CodePipeline Détecte d'abord les modifications apportées à l'artefact source.



CodePipeline envoie le code mis à jour vers votre pile dans AWS OpsWorks Stacks.



12. Lorsque les deux étapes du pipeline sont terminées avec succès, ouvrez votre pile dans AWS OpsWorks Stacks (MyStack).
13. Sur la page MyStackdes propriétés, sélectionnez Instances.
14. Dans la colonne IP publique, sélectionnez l'adresse IP publique de votre instance pour afficher le texte de l'application mise à jour.



Étape 7 (facultatif) : Nettoyer les ressources

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Pour éviter des frais indésirables sur votre compte AWS, vous pouvez supprimer les AWS ressources que vous avez utilisées pour cette procédure pas à pas. Ces AWS ressources incluent la pile AWS OpsWorks Stacks, le rôle IAM et le profil d'instance, ainsi que le pipeline dans lequel vous avez créé. CodePipeline Cependant, vous souhaiterez peut-être continuer à utiliser ces AWS ressources pour en savoir plus sur AWS OpsWorks Stacks et CodePipeline. Si vous souhaitez conserver ces ressources, vous avez terminé cette procédure pas à pas.

Pour supprimer l'application de la pile

Comme vous n'avez ni créé ni appliqué l'application dans le cadre de votre AWS CloudFormation modèle, supprimez l'application de test PHP avant de supprimer le stack in AWS CloudFormation.

1. Dans la console AWS OpsWorks Stacks, dans le panneau de navigation des services, choisissez Applications.
2. Sur la page Applications, sélectionnez PHPTestApplication, puis dans Actions, choisissez Supprimer. Lorsque vous êtes invité à confirmer la suppression, choisissez Delete (Supprimer). AWS OpsWorks Stacks supprimera l'application.

Pour supprimer la pile

Comme vous avez créé la pile en exécutant un AWS CloudFormation modèle, vous pouvez supprimer la pile, y compris la couche, l'instance, le profil d'instance et le groupe de sécurité créés par le modèle, dans la AWS CloudFormation console.

1. Ouvrez la AWS CloudFormation console.

2. Dans le tableau de bord de la AWS CloudFormation console, sélectionnez la pile que vous avez créée (MyStack). Dans le menu Actions , sélectionnez Delete Stack (Supprimer la pile). Lorsque vous êtes invité à confirmer, choisissez Oui, supprimer.
3. Attendez que DELETE_COMPLETE apparaisse dans la colonne Statut de la pile.

Pour supprimer le pipeline

1. Ouvrez la CodePipeline console.
2. Dans le CodePipeline tableau de bord, choisissez le pipeline que vous avez créé pour cette procédure.
3. Sur la page du pipeline, choisissez Modifier.
4. Sur la page Edit, choisissez Delete. Lorsque vous êtes invité à confirmer, choisissez Delete.

Utilisation de la AWS OpsWorks CLI Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

L'interface de ligne de commande (CLI) AWS OpsWorks Stacks fournit les mêmes fonctionnalités que la console et peut être utilisée pour diverses tâches. La CLI AWS OpsWorks Stacks fait partie du AWS CLI. Pour plus d'informations, notamment sur la façon d'installer et de configurer le AWS CLI, consultez Qu'est-ce que l'interface de ligne de commande AWS ? . Pour obtenir une description complète de chaque commande, consultez la Référence AWS OpsWorks Stacks.

> **Note**
>
> Si vous utilisez un poste de travail Windows, vous pouvez également exécuter les AWS Outils pour Windows PowerShell afin d'effectuer des opérations AWS OpsWorks Stacks à

partir de la ligne de commande. Pour plus d'informations, consultez la section Outils AWS pour Windows PowerShell.

AWS OpsWorks Les commandes Stacks ont le format général suivant :

```
aws opsworks --region us-west-1 opsworks command-name [--argument1 value] [...]
```

Si une valeur d'argument est un objet JSON, vous devez faire précéder d'une séquence d'échappement les caractères ", sans quoi la commande peut retourner une erreur selon laquelle le JSON n'est pas valide. Par exemple, si l'objet JSON est `"{"somekey":"somevalue"}"`, il doit être se présenter sous la forme `"{\"somekey\":\"somevalue\"}"`. Une autre approche consiste à placer l'objet JSON dans un fichier et à utiliser `file://` pour l'inclure dans la ligne de commande. L'exemple suivant crée une application à l'aide d'un objet source application stocké dans appsource.json.

```
aws opsworks --region us-west-1 create-app --stack-id 8c428b08-a1a1-46ce-a5f8-
feddc43771b8 --name SimpleJSP --type java --app-source file://appsource.json
```

La plupart des commandes renvoient une ou plusieurs valeurs, empaquetées en tant qu'objet JSON. Les sections suivantes contiennent des exemples. Pour une description détaillée des valeurs retournées par chaque commande, consultez la Référence AWS OpsWorks Stacks.

> **Note**
>
> AWS CLI les commandes doivent spécifier une région, comme indiqué dans les exemples. Les valeurs valides du paramètre --region sont affichées dans le tableau suivant. Pour simplifier vos chaînes de commande AWS OpsWorks Stacks, configurez la CLI pour spécifier votre région par défaut, afin de pouvoir omettre le `--region` paramètre. Si vous travaillez généralement sur plusieurs points de terminaison régionaux, ne configurez pas le AWS CLI pour utiliser un point de terminaison régional par défaut. Le point de terminaison de la région du Canada (Centre) est AWS CLI uniquement disponible dans l'API ; il n'est pas disponible pour les piles que vous créez dans le AWS Management Console. Pour plus d'informations, consultez Configuration de la région AWS.

Nom de la région	Code de commande
Région USA Est (Ohio)	us-east-2
Région USA Est (Virginie du Nord)	us-east-1
Région US West (N. California)	us-west-1
Région USA Ouest (Oregon)	us-west-2
Région Canada (Centre)	ca-central-1
Région Europe (Irlande)	eu-west-1
Région Europe (Londres)	eu-west-2
Région Europe (Paris)	eu-west-3
Région Europe (Francfort)	eu-central-1
Région Asie-Pacifique (Tokyo)	ap-northeast-1
Région Asie-Pacifique (Séoul)	ap-northeast-2
Région Asie-Pacifique (Mumbai)	ap-south-1
Région Asie-Pacifique (Singapour)	ap-southeast-1
Région Asie-Pacifique (Sydney)	ap-southeast-2
Région Amérique du Sud (São Paulo)	sa-east-1

Pour utiliser une interface de ligne commande, vous devez avoir les autorisations appropriées. Pour plus d'informations sur les autorisations AWS OpsWorks Stacks, consultez Gestion des autorisations utilisateur. Pour déterminer les autorisations nécessaires pour une commande particulière, consultez la page de référence de la commande dans la Référence AWS OpsWorks Stacks.

Les sections suivantes décrivent comment utiliser la CLI AWS OpsWorks Stacks pour effectuer diverses tâches courantes.

Créer une instance (create-instance)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Utilisez la commande create-instance pour créer une instance dans une pile spécifiée.

Rubriques

- Créer une instance avec un nom d'hôte par défaut
- Créer une instance avec un nom d'hôte à thème
- Créer une instance avec une AMI personnalisée

Créer une instance avec un nom d'hôte par défaut

```
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"
```

Les arguments sont les suivants :

- `stack-id`— Vous pouvez obtenir l'ID de pile sur la page des paramètres de la pile sur la console (recherchez l'OpsWorks ID) ou en appelant describe-stacks.
- `layer-ids`— Vous pouvez obtenir une couche IDs depuis la page de détails de la couche sur la console (recherchez l'OpsWorks ID) ou en appelant describe-layers. Dans cet exemple, l'instance appartient à une seule couche.
- `instance-type`— La spécification qui définit la mémoire, le processeur, la capacité de stockage et le coût horaire de l'instance, dans cet `m1.large` exemple.
- `os`— Le système d'exploitation de l'instance ; Amazon Linux pour cet exemple.

La commande renvoie un objet JSON qui contient l'ID d'instance, comme suit :

```
{
    "InstanceId": "5f9adeaa-c94c-42c6-aeef-28a5376002cd"
}
```

Cet exemple crée une instance avec un nom d'hôte par défaut, qui est simplement un nombre entier. La section suivante décrit comment créer une instance avec un nom d'hôte généré à partir d'un thème.

Créer une instance avec un nom d'hôte à thème

Vous pouvez également créer une instance avec un nom d'hôte à thème. Vous spécifiez le thème lorsque vous créez la pile. Pour plus d'informations, consultez Créer une pile .Pour créer l'instance, appelez d'abord get-hostname-suggestionpour générer un nom. Par exemple :

```
C:\>aws opsworks get-hostname-suggestion --region us-west-1 --layer-id 5c8c272a-
f2d5-42e3-8245-5bf3927cb65b
```

Si vous spécifiez le thème par défaut `Layer Dependent`, `get-hostname-suggestion` ajoute simplement un chiffre au nom court de la couche. Pour de plus amples informations, veuillez consulter Créer une pile.

La commande renvoie le nom d'hôte généré.

```
{
    "Hostname": "php-app2",
    "LayerId": "5c8c272a-f2d5-42e3-8245-5bf3927cb65b"
}
```

Vous pouvez ensuite utiliser l'argument `hostname`pour transmettre le nom généré à `create-instance`, comme suit :

```
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"
```

Créer une instance avec une AMI personnalisée

La commande create-instance suivante crée une instance avec une AMI personnalisée, qui doit provenir de la région de la pile. Pour plus d'informations sur la création d'une AMI personnalisée pour AWS OpsWorks Stacks, consultezUtilisation de Custom AMIs.

```
C:\>aws opsworks create-instance --region us-west-1 --stack-id c5ef46ce-3ccd-472c-
a3de-9bec94c6028e
   --layer-ids 6ff8a2ac-c9cc-49cf-9c67-fc852539ade4 --instance-type c3.large --os
 Custom
   --ami-id ami-6c61f104
```

Les arguments sont les suivants :

- `stack-id`— Vous pouvez obtenir l'ID de pile sur la page des paramètres de la pile sur la console (recherchez l'OpsWorks ID) ou en appelant describe-stacks.
- `layer-ids`— Vous pouvez obtenir une couche IDs depuis la page de détails de la couche sur la console (recherchez l'OpsWorks ID) ou en appelant describe-layers. Dans cet exemple, l'instance appartient à une seule couche.
- `instance-type`— La valeur définit la mémoire, le processeur, la capacité de stockage et le coût horaire de l'instance, et doit être compatible avec l'AMI (dans cet `c3.large` exemple).
- `os`— Le système d'exploitation de l'instance, qui doit être configuré `Custom` pour une AMI personnalisée.
- `ami-id`— L'identifiant de l'AMI, qui devrait ressembler à `ami-6c61f104`

> **Note**
>
> Lorsque vous utilisez une AMI personnalisée, les mappages de périphérique de stockage en mode bloc ne sont pas pris en charge et les valeurs que vous spécifiez pour l'option `--block-device-mappings` sont ignorées.

La commande renvoie un objet JSON qui contient l'ID d'instance, comme suit :

```
{
```

```
    "InstanceId": "5f9adeaa-c94c-42c6-aeef-28a5376002cd"
}
```

Déployer une application (create-deployment)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Utilisez la commande create-deployment pour déployer une application sur une pile spécifiée.

Rubriques

- Déployer une application

Déployer une application

```
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\"}"
```

Les arguments sont les suivants :

- `stack-id`— Vous pouvez obtenir l'ID de pile sur la page des paramètres de la pile sur la console (recherchez l'OpsWorks ID) ou en appelant`describe-stacks`.
- `app-id`— Vous pouvez obtenir l'identifiant de l'application sur la page de détails de l'application (recherchez l'OpsWorks identifiant) ou en appelant describe-apps.
- `command`— L'argument prend un objet JSON qui définit le nom de la commande sur`deploy`, qui déploie l'application spécifiée dans la pile.

  Notez que les caractères " de l'objet JSON sont tous précédés d'une séquence d'échappement. Sinon, la commande peut retourner une erreur selon laquelle le JSON n'est pas valide.

La commande renvoie un objet JSON qui contient l'ID de déploiement, comme suit :

```
{
    "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"
}
```

> **Note**
>
> L'exemple précédent déploie sur chaque instance de la pile. Pour effectuer un déploiement sur un sous-ensemble d'instances spécifié, ajoutez un `instance-ids` argument et listez l'instance IDs.

Afficher les applications d'une pile (describe-apps)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Utilisez la commande describe-apps pour répertorier les applications d'une pile ou obtenir des détails sur les applications spécifiées.

```
aws opsworks --region us-west-1 describe-apps --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

L'exemple précédent retourne un objet JSON qui contient des informations sur chaque application. Cet exemple n'a qu'une seule application. Pour obtenir une description de chaque paramètre, consultez 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"
    }
  ]
}
```

Afficher les commandes d'une pile (describe-commands)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Utilisez la commande describe-commands pour répertorier les commandes d'une pile ou obtenir des détails sur les commandes spécifiées. L'exemple suivant obtient les informations sur les commandes qui ont été exécutées sur une instance spécifiée.

```
aws opsworks --region us-west-1 describe-commands --instance-id
 8c2673b9-3fe5-420d-9cfa-78d875ee7687
```

La commande renvoie un objet JSON qui contient les détails relatifs à chaque commande. Le paramètre Type identifie le nom de la commande, deploy ou undeploy dans cet exemple. Pour obtenir une description des autres paramètres, consultez 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
    }
  ]
}
```

Afficher les déploiements d'une pile (describe-deployments)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Utilisez la commande describe-deployments pour répertorier les déploiements d'une pile ou obtenir des détails sur les déploiements spécifiés.

```
aws opsworks --region us-west-1 describe-deployments --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a
```

La commande précédente retourne un objet JSON qui contient les détails relatifs à chaque déploiement de la pile spécifiée. Pour obtenir une description de chaque paramètre, consultez 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"
        }
    ]
}
```

Répertorier les adresses IP élastiques d'une pile (describe-elastic-ips)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Utilisez la commande describe-elastic-ips pour afficher les adresses IP Elastic qui ont été enregistrées auprès d'une pile ou obtenir des détails sur les adresses IP Elastic spécifiées.

```
aws opsworks --region us-west-2 describe-elastic-ips --instance-id b62f3e04-e9eb-436c-a91f-d9e9a396b7b0
```

La commande précédente retourne un objet JSON qui contient les détails relatifs à chaque adresse IP Elastic (une dans cet exemple) pour une instance spécifiée. Pour obtenir une description de chaque paramètre, consultez describe-elastic-ips.

```
{
```

```
  "ElasticIps": [
      {
          "Ip": "192.0.2.0",
          "Domain": "standard",
          "Region": "us-west-2"
      }
  ]
}
```

Afficher les instances d'une pile (describe-instances)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Utilisez la commande describe-instances pour répertorier les instances d'une pile ou obtenir des détails sur les instances spécifiées.

```
C:\>aws opsworks --region us-west-2 describe-instances --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

La commande précédente retourne un objet JSON qui contient des détails sur chaque instance d'une pile spécifiée. Pour obtenir une description de chaque paramètre, consultez 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"
        }
    ]
}
```


Afficher les piles d'un compte (describe-stacks)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Utilisez la commande describe-stacks pour répertorier les piles d'un compte ou obtenir des détails sur les piles spécifiées.

```
aws opsworks --region us-west-2 describe-stacks
```

La commande précédente retourne un objet JSON qui contient des détails sur chaque pile du compte, deux dans cet exemple. Pour obtenir une description de chaque paramètre, consultez 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"
        }
    ]
}
```

Afficher les couches d'une pile (describe-layers)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Utilisez la commande describe-layers pour répertorier les couches d'une pile ou obtenir des détails sur les couches spécifiées.

```
aws opsworks --region us-west-2 describe-layers --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

La commande précédente renvoie un objet JSON qui contient des détails sur chaque couche d'une pile spécifiée, dans cet exemple, une couche MySQL et une couche personnalisée. Pour obtenir une description de chaque paramètre, consultez 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": []
    }
  ]
}
```

Exécuter une recette (create-deployment)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Utilisez la commande create-deployment pour exécuter les commandes de pile et les commandes de déploiement. L'exemple suivant exécute une commande de pile pour exécuter une recette personnalisée sur une pile spécifiée.

```
aws opsworks --region us-west-1 create-deployment --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb
    --command "{\"Name\":\"execute_recipes\", \"Args\":{\"recipes\":[\"phpapp::appsetup
\"]}}"
```

L'argument `command` accepte un objet JSON mis en forme comme suit :

- `Name` : spécifie le nom de la commande. La commande `execute_recipes` utilisée pour cet exemple exécute une recette spécifiée sur les instances de la pile.
- `Args` : spécifie une liste d'arguments et leurs valeurs. Cet exemple possède un seul argument, `recipes`, qui est défini sur la recette à exécuter, `phpapp::appsetup`.

Notez que les caractères " de l'objet JSON sont tous précédés d'une séquence d'échappement. Sinon, la commande peut retourner une erreur selon laquelle le JSON n'est pas valide.

La commande renvoie un ID de déploiement, que vous pouvez utiliser afin d'identifier la commande pour d'autres commandes d'interface de ligne de commande, telles que `describe-commands`.

```
{
    "DeploymentId": "5cbaa7b9-4e09-4e53-aa1b-314fbd106038"
}
```

Installer les dépendances (create-deployment)

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si

vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Utilisez la commande create-deployment pour exécuter les commandes de pile et les commandes de déploiement. L'exemple suivant exécute la commande de pile `update_dependencies` pour mettre à jour les dépendances sur les instances d'une pile.

```
aws opsworks --region us-west-1 create-deployment --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb
--command "{\"Name\":\"install_dependencies\"}"
```

L'argument `command` accepte un objet JSON avec un paramètre `Name` dont la valeur spécifie le nom de la commande, `install_dependencies` dans cet exemple. Notez que les caractères " de l'objet JSON sont tous précédés d'une séquence d'échappement. Sinon, la commande peut retourner une erreur selon laquelle le JSON n'est pas valide.

La commande renvoie un ID de déploiement, que vous pouvez utiliser afin d'identifier la commande pour d'autres commandes d'interface de ligne de commande, telles que `describe-commands`.

```
{
    "DeploymentId": "aef5b255-8604-4928-81b3-9b0187f962ff"
}
```

Mettre à jour la configuration de pile (update-stack)

Important

Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Utilisez la commande update-stack pour mettre à jour la configuration d'une pile spécifiée. L'exemple suivant met à jour une pile pour ajouter le JSON personnalisé aux attributs de configuration de la pile.

```
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
```

Notez que les caractères " de l'objet JSON sont tous précédés d'une séquence d'échappement. Sinon, la commande peut retourner une erreur selon laquelle le JSON n'est pas valide.

> **Note**
>
> L'exemple spécifie également un rôle de service pour la pile. Vous devez définir `service-role-arn` avec un ARN de rôle de service valide, sans quoi l'action échoue ; il n'y a aucune valeur par défaut. Vous pouvez spécifier l'ARN du rôle de service actuel de la pile, si vous préférez, mais vous devez le faire explicitement.

La commande `update-stack` ne retourne aucune valeur.

Guide de débogage et dépannage

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé pour les nouveaux clients et les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Si vous avez besoin de déboguer une recette ou de résoudre un problème de service, la meilleure approche consiste généralement à exécuter les étapes suivantes, dans l'ordre :

1. Recherchez votre problème dans Débogage et dépannage des problèmes courants.
2. Recherchez sur le forum AWS OpsWorks Stacks pour voir si le problème a déjà été abordé.

   Le forum comprend de nombreux utilisateurs expérimentés et est surveillé par l'équipe AWS OpsWorks Stacks.

3. Pour tout problème avec les recettes, consultez Débogage des recettes.
4. Contactez l'assistance AWS OpsWorks Stacks ou publiez votre problème sur le forum AWS OpsWorks Stacks.

La section suivante propose des conseils pour le débogage des recettes. La dernière section décrit les problèmes courants de débogage et de dépannage, et leurs solutions.

> **Note**
>
> Chaque exécution de Chef produit un journal qui fournit une description détaillée de l'exécution et constitue une ressource précieuse de dépannage. Pour spécifier la quantité de détails dans le journal, ajoutez une instruction `Chef::Log.level` pour une recette personnalisée qui spécifie le niveau de journalisation souhaité. La valeur par défaut est `:info`. L'exemple suivant montre comment définir le niveau de journalisation de Chef sur `:debug` afin de fournir la description la plus détaillée de l'exécution.
>
> ```
> Chef::Log.level = :debug
> ```
>
> Pour plus d'informations sur l'affichage et l'interprétation des journaux de Chef, consultez Journaux de Chef.

Rubriques

- Débogage des recettes
- Débogage et dépannage des problèmes courants

Débogage des recettes

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Lorsqu'un événement du cycle de vie se produit ou si vous exécutez la commande de pile Execute Recipes, AWS OpsWorks Stacks émet une commande auprès de l'agent pour lancer une exécution Chef Solo sur les instances spécifiées afin d'exécuter les recettes appropriées, y compris vos recettes personnalisées. Cette section décrit quelques méthodes de débogage des recettes ayant échoué.

Rubriques

- Connexion à une instance ayant échoué
- Journaux de Chef
- Utilisation de la CLI AWS OpsWorks de l'agent Stacks

Connexion à une instance ayant échoué

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Si une recette échoue, l'instance n'est pas en ligne et son état devient `setup_failed`. Même si l'instance n'est pas en ligne en ce qui concerne AWS OpsWorks Stacks, elle est en cours d'exécution et il est souvent utile de se connecter pour résoudre le problème. EC2 Par exemple, vous pouvez vérifier si une application ou un livre personnalisé est correctement installé. Le support intégré de AWS OpsWorks Stacks pour la connexion SSH et RDP n'est disponible que pour les instances en ligne. Toutefois, si vous avez attribué une paire de clés SSH à l'instance, vous pouvez toujours vous connecter, comme suit :

- Instances Linux : utilisez la clé privée de la paire de clés SSH pour vous connecter à un client SSH tiers, tel qu'OpenSSH ou PuTTY.

  Vous pouvez utiliser une paire de EC2 clés ou votre paire de clés SSH personnelle à cette fin.
- Instances Windows : utilisez la EC2 clé privée de la paire de clés pour récupérer le mot de passe administrateur de l'instance.

Utilisez ce mot de passe pour vous connecter avec votre client RDP préféré. Pour de plus amples informations, veuillez consulter Connexion en tant qu'Administrateur. Vous ne pouvez pas utiliser de paire de clés SSH personnelle pour récupérer un mot de passe administrateur.

Journaux de Chef

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les journaux Chef sont l'une de vos principales ressources de dépannage, en particulier pour le débogage de recettes. AWS OpsWorks Stacks capture le journal Chef pour chaque commande et conserve les journaux des 30 commandes les plus récentes d'une instance. Etant donné que l'exécution est en mode débogage, le journal contient une description détaillée de l'exécution de Chef, y compris le texte qui est envoyé à `stdout` et `stderror`. Si une recette échoue, le journal inclut la trace de la pile de Chef.

AWS OpsWorks Stacks vous propose plusieurs façons de consulter les journaux Chef. Une fois que vous disposez des informations du journal, vous pouvez les utiliser pour déboguer les recettes ayant échoué.

> **Note**
>
> Vous pouvez aussi afficher la queue d'un journal spécifié en utilisant SSH pour vous connecter à l'instance et en exécutant la commande `show_log` de l'interface de ligne de commande de l'agent. Pour de plus amples informations, veuillez consulter Affichage des journaux de Chef.

Rubriques

- Affichage d'un journal de Chef avec la console

- Affichage d'un journal de Chef avec l'API ou l'interface de ligne de commande
- Affichage d'un journal de Chef sur une instance
- Interprétation d'un journal Chef
- Erreurs courantes dans le journal de Chef

Affichage d'un journal de Chef avec la console

La façon la plus simple pour afficher un journal de Chef consiste à accéder à la page des détails de l'instance. La section Logs inclut une entrée pour chaque événement et la commande Execute Recipes. Voici la section Logs d'une instance avec les commandes configure et setup qui correspondent à aux événements de cycle de vie Configure et Setup.



Cliquez sur show dans la colonne Log de la commande appropriée pour afficher le journal de Chef correspondant. En cas d'erreur, AWS OpsWorks Stacks ouvre automatiquement le journal de l'erreur, qui se trouve généralement à la fin du fichier.

Affichage d'un journal de Chef avec l'API ou l'interface de ligne de commande

Vous pouvez utiliser la `describe-commands`commande AWS OpsWorks Stacks CLI ou l'action `DescribeCommands`API pour consulter les journaux, qui sont stockés dans un compartiment Amazon S3. Voici comment utiliser l'interface de ligne de commande pour afficher l'un des fichiers journaux en cours pour une instance spécifiée. La procédure d'utilisation de `DescribeCommands` est globalement similaire.

Pour utiliser les AWS OpsWorks Stacks pour afficher les journaux Chef d'une instance

1. Ouvrez la page de détails de l'instance et copiez sa valeur OpsWorksd'ID.
2. Utilisez la valeur d'ID pour exécuter la commande d'interface de ligne de commande `describe-commands`, comme suit :

```
aws opsworks describe-commands --instance-id 67bf0da2-29ed-4217-990c-d895d51812b9
```

La commande renvoie un objet JSON avec un objet intégré pour chaque commande exécutée par AWS OpsWorks Stacks sur l'instance, la plus récente en premier. Le paramètre `Type` contient le type de commande pour chaque objet intégré, une commande `configure` et une commande `setup` dans cet exemple.


```
{
    "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. Copiez la valeur `LogUrl` dans votre navigateur pour afficher le journal.

Si l'instance a plus de quelques commandes, vous pouvez ajouter des paramètres à `describe-commands` pour filtrer les commandes à inclure dans l'objet de réponse. Pour plus d'informations, consultez describe-commands.

Affichage d'un journal de Chef sur une instance

> **Note**
>
> Les rubriques de cette section s'appliquent à Chef 12. Pour plus d'informations sur l'emplacement des journaux de Chef pour Chef 11.10 et les versions antérieures, consultez Résolution des problèmes de Chef 11.10 et des versions antérieures pour Linux.

Instances Linux

AWS OpsWorks Stacks stocke les journaux Chef de chaque instance dans son `/var/chef/runs` répertoire. Pour les instances Linux, ce répertoire inclut également les conteneurs de données associés, stockés sous forme de fichiers au format JSON. Vous avez besoin de privilèges sudo pour accéder à ce répertoire. Le journal de chaque exécution se trouve dans un fichier nommé `chef.log` dans le sous-répertoire de l'exécution concernée.

AWS OpsWorks Stacks stocke ses journaux internes dans le `/var/log/aws/opsworks` dossier de l'instance. Les informations sont rarement utiles pour le dépannage. Cependant, ces journaux sont utiles au support de AWS OpsWorks Stacks, et il se peut que l'on vous demande de les fournir si vous rencontrez un problème avec le service. Les journaux Linux peuvent également fournir des données de dépannage utiles dans certains cas.

instances Windows

Journaux de l'agent

Sur les instances Windows, les OpsWorks journaux sont stockés dans un `ProgramData` chemin tel que le suivant. Le nombre inclut un horodatage.

```
C:\ProgramData\OpsWorksAgent\var\logs\number
```

> **Note**
>
> Par défaut, `ProgramData` est un dossier masqué. Pour l'afficher, accédez à Folder Options (Options du dossier). Sous Afficher, sélectionnez l'option d'affichage des fichiers masqués.

L'exemple suivant montre les journaux de l'agent sur une instance Windows.

```
Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---          5/24/2015  11:59 PM     127277 command.20150524.txt
-a---          5/25/2015  11:59 PM     546772 command.20150525.txt
-a---          5/26/2015  11:59 PM     551514 command.20150526.txt
-a---          5/27/2015   9:43 PM     495181 command.20150527.txt
-a---          5/24/2015  11:59 PM      24353 keepalive.20150524.txt
-a---          5/25/2015  11:59 PM     106232 keepalive.20150525.txt
-a---          5/26/2015  11:59 PM     106208 keepalive.20150526.txt
-a---          5/27/2015   8:54 PM      92593 keepalive.20150527.txt
-a---          5/24/2015   7:19 PM       3891 service.20150524.txt
-a---          5/27/2015   8:54 PM       1493 service.20150527.txt
-a---          5/24/2015  11:59 PM     112549 wire.20150524.txt
-a---          5/25/2015  11:59 PM     501501 wire.20150525.txt
-a---          5/26/2015  11:59 PM     499640 wire.20150526.txt
-a---          5/27/2015   8:54 PM     436870 wire.20150527.txt
```

Journaux de Chef

Pour les instances Windows, les journaux de Chef sont stockés dans un chemin `ProgramData` comme suit. Le nombre inclut un horodatage.

```
C:\ProgramData\OpsWorksAgent\var\commands\number
```

> **Note**
>
> Ce répertoire contient uniquement la sortie de la première exécution (OpsWorks possédée) de Chef.

L'exemple suivant montre les journaux Chef OpsWorks détenus sur une instance Windows.

```
 Mode                 LastWriteTime            Name
 ----                 -------------            ----
 d----         5/24/2015   7:23 PM
 configure-7ecb5f47-7626-439b-877f-5e7cb40ab8be
 d----         5/26/2015   8:30 PM            configure-8e74223b-d15d-4372-aeea-
a87b428ffc2b
 d----         5/24/2015   6:34 PM            configure-
c3980a1c-3d08-46eb-9bae-63514cee194b
 d----         5/26/2015   8:32 PM            grant_remote_access-70dbf834-1bfa-4fce-
b195-e50e85402f4c
 d----         5/26/2015  10:30 PM            revoke_remote_access-1111fce9-843a-4b27-
b93f-ecc7c5e9e05b
 d----         5/24/2015   7:21 PM            setup-754ec063-8b60-4cd4-
b6d7-0e89d7b7aa78
 d----         5/26/2015   8:27 PM            setup-af5bed36-5afd-4115-
af35-5766f88bc039
 d----         5/24/2015   6:32 PM            setup-d8abeffa-24d4-414b-
bfb1-4ad07319f358
 d----         5/24/2015   7:13 PM            shutdown-c7130435-9b5c-4a95-
be17-6b988fc6cf9a
 d----         5/26/2015   8:25 PM
 sync_remote_users-64c79bdc-1f6f-4517-865b-23d2def4180c
 d----         5/26/2015   8:48 PM
 update_custom_cookbooks-2cc59a94-315b-414d-85eb-2bdea6d76c6a
```

Journaux Chef utilisateur

Les journaux de vos exécutions de Chef sont disponibles dans les fichiers nommés `logfile.txt` dans un dossier nommé conformément à la commande Chef numérotée, comme illustré ci-après.

C : /chef `runs` `command-12345` `attribs.json` `client.rb` `logfile.txt`

Interprétation d'un journal Chef

Le début du journal contient principalement la journalisation interne de Chef.

```
# 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
...
```

Cette partie du fichier est utile essentiellement pour les experts de Chef. Notez que la liste d'exécution inclut uniquement deux recettes, même si la plupart des commandes en impliquent bien plus. Ces deux recettes gèrent la tâche de chargement et d'exécution de toutes les autres recettes intégrées et personnalisées.

La partie la plus intéressante du fichier est généralement à la fin. Si une exécution se termine avec succès, vous devriez voir quelque chose de similaire à ce qui suit :

```
...
[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**
>
> Vous pouvez utiliser l'interface de ligne de commande de l'agent pour afficher la queue du journal pendant ou après l'exécution. Pour de plus amples informations, veuillez consulter Affichage des journaux de Chef.

En cas d'échec d'une recette, vous devez rechercher une sortie de niveau ERROR, qui contient une exception suivie d'une trace de la pile de Chef, comme indiqué ci-après :

```
...
Please report any problems with the /usr/scripts/mysqlbug script!

[  OK  ]
MySQL Daemon failed to start.
Starting mysqld:  [FAILED]STDERR: 130611 15:07:55 [Warning] The syntax '--log-slow-
queries' is deprecated and will be removed in a future release. Please use '--slow-
query-log'/'--slow-query-log-file' instead.
130611 15:07:56 [Warning] The syntax '--log-slow-queries' is deprecated and will be
 removed in a future release. Please use '--slow-query-log'/'--slow-query-log-file'
 instead.
---- End output of /sbin/service mysqld start ----

/opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/mixin/command.rb:184:in `handle_command_failures'
```

```
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/mixin/command.rb:131:in `run_command'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/provider/service/init.rb:37:in `start_service'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/provider/service.rb:60:in `action_start'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource.rb:406:in `send'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource.rb:406:in `run_action'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/runner.rb:53:in `run_action'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/runner.rb:89:in `converge'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/runner.rb:89:in `each'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/runner.rb:89:in `converge'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource_collection.rb:94:in `execute_each_resource'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource_collection/stepable_iterator.rb:116:in `call'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource_collection/stepable_iterator.rb:116:in
 `call_iterator_block'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource_collection/stepable_iterator.rb:85:in `step'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource_collection/stepable_iterator.rb:104:in `iterate'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource_collection/stepable_iterator.rb:55:in
 `each_with_index'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/resource_collection.rb:92:in `execute_each_resource'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/runner.rb:84:in `converge'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/client.rb:268:in `converge'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/client.rb:158:in `run'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/application/solo.rb:190:in `run_application'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/application/solo.rb:181:in `loop'
```

```
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/application/solo.rb:181:in `run_application'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/../lib/chef/application.rb:62:in `run'
  /opt/aws/opsworks/releases/20130605160141_122/vendor/bundle/ruby/1.8/gems/
chef-0.9.15.5/bin/chef-solo:25
  /opt/aws/opsworks/current/bin/chef-solo:16:in `load'
  /opt/aws/opsworks/current/bin/chef-solo:16
```

La fin du fichier est la trace de la pile de Chef. Vous devez également examiner la sortie juste avant l'exception, car elle contient souvent une erreur système comme `package not available` qui peut également être utile pour déterminer la cause de l'échec. Dans ce cas, le daemon MySQL n'a pas pu démarrer.

Erreurs courantes dans le journal de Chef

Voici quelques erreurs courantes qui apparaissent dans le journal de Chef et comment les traiter.

Impossible de trouver le journal

Au début d'une exécution de Chef, les instances reçoivent une URL Amazon S3 présignée qui vous permet de consulter le journal sur une page Web lorsque l'exécution de Chef est terminée. Comme cette URL expire au bout de deux heures, aucun journal n'est chargé sur le site Amazon S3 si une exécution de Chef prend plus de deux heures, même si aucun problème n'est survenu pendant l'exécution de Chef. La commande de création d'un journal a réussi, mais le journal peut être vu uniquement sur l'instance, pas au niveau de l'URL présignée.

Fin brutale du journal

Si un journal de Chef se termine brutalement sans indiquer la réussite ni afficher des informations d'erreur, vous avez probablement rencontré un état de faible mémoire qui a empêché Chef de remplir le journal. La meilleure solution consiste à essayer de nouveau avec une plus grande instance.

Livre de recettes ou recette manquant

Si l'exécution de Chef doit gérer un livre de recettes ou une recette qui ne sont pas dans le cache du livre de recettes, vous verrez quelque chose de similaire à ce qui suit :

```
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?
```

Cette entrée indique que le livre de recettes `mycookbook` n'est pas dans le cache de livre de recettes. Avec Chef 11.4, vous pouvez également rencontrer cette erreur si vous ne déclarez pas les dépendances correctement dans `metadata.rb`.

AWS OpsWorks Stacks exécute des recettes à partir du cache du livre de recettes de l'instance. Il télécharge les livres de recettes du référentiel vers le cache lorsque l'instance démarre. Cependant, AWS OpsWorks Stacks ne met pas automatiquement à jour le cache d'une instance en ligne si vous modifiez ultérieurement les livres de recettes de votre référentiel. Si vous avez modifié vos livres de recettes ou si vous en avez ajouté de nouveaux depuis le début de l'instance, procédez comme suit :

1. Assurez-vous que vous avez validé vos modifications dans le référentiel.
2. Exécutez la commande de pile Update Cookbooks (Mettre à jour les livres de recettes) pour mettre à jour le cache du livre de recettes avec la version la plus récente du référentiel.

Échec de la commande locale

Si une ressource `execute` de Chef ne parvient pas à exécuter la commande spécifiée, vous verrez quelque chose de similaire à ce qui suit :

```
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=/
```

Faites défiler le journal vers le haut et vous devriez voir la sortie `stderr` et `stdout` de la commande, ce qui vous aidera à déterminer pourquoi la commande a échoué.

Échec du package

En cas d'échec d'une installation de package, vous verrez quelque chose de similaire à ce qui suit :

```
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
```

Faites défiler le journal vers le haut et vous devriez voir la sortie STDOUT et STDERROR de la commande, ce qui vous aidera à déterminer pourquoi l'installation du package a échoué.

Utilisation de la CLI AWS OpsWorks de l'agent Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> L'interface de ligne de commande de l'agent est disponible uniquement sur les instances Linux.

Sur chaque instance en ligne, AWS OpsWorks Stacks installe un agent qui communique avec le service. Le service AWS OpsWorks Stacks envoie à son tour des commandes à l'agent pour effectuer des tâches telles que le lancement des exécutions de Chef sur l'instance lorsqu'un événement du cycle de vie se produit. Sur les instances Linux, l'agent expose une interface de ligne de commande (CLI) qui est très utile pour le dépannage. Pour exécuter les commandes de l'interface de ligne de commande de l'agent, utilisez SSH pour vous connecter à une instance. Vous pouvez ensuite exécuter les commandes de l'interface de ligne de commande de l'agent pour effectuer diverses tâches, y compris les suivantes :

- Exécuter les recettes.
- Afficher les journaux de Chef.
- Afficher le JSON de la configuration et du déploiement de la pile.

Pour plus d'informations sur la configuration d'une connexion SSH à une instance, consultez Connexion avec SSH. Vous devez également disposer d'autorisations SSH et sudo pour la pile.

Cette section décrit comment utiliser l'interface de ligne de commande de l'agent pour le dépannage. Pour plus d'informations et une référence complète sur les commandes, consultez AWS OpsWorks CLI de l'agent Stacks.

Rubriques

- Exécution des recettes
- Affichage des journaux de Chef
- Affichage du JSON de configuration et de déploiement de la pile

Exécution des recettes

La commande `run_command` de l'interface de ligne de commande de l'agent demande à l'agent d'exécuter à nouveau une commande qu'il a déjà effectuée. Les commandes les plus utiles pour le dépannage, `setup`, `configure`, `deploy` et `undeploy`, correspondent chacune à un événement du cycle de vie. Elles demandent à l'agent de lancer une exécution de Chef afin d'exécuter les recettes associées.

> **Note**
>
> La commande `run_command` est limitée à l'exécution du groupe de recettes qui est associé à une commande spécifique, généralement les recettes qui sont associées à un événement du cycle de vie. Vous ne pouvez pas l'utiliser pour exécuter une recette particulière. Pour exécuter une ou plusieurs recettes spécifiées, utilisez la commande de pile Execute Recipes ou les actions équivalentes de l'API ou de l'interface de ligne de commande (`create-deployment` et `CreateDeployment`).

La commande `run_command` est très utile pour le débogage des recettes personnalisées, en particulier les recettes qui sont affectées aux événements du cycle de vie Setup et Configure, que vous ne pouvez pas déclencher directement à partir de la console. En utilisant `run_command`, vous pouvez exécuter les recettes d'un événement particulier aussi souvent que nécessaire sans avoir à démarrer ou interrompre les instances.

> **Note**
>
> AWS OpsWorks Stacks exécute les recettes à partir du cache du livre de recettes de l'instance, et non du référentiel de livres de recettes. AWS OpsWorks Stacks télécharge

les livres de recettes dans ce cache au démarrage de l'instance, mais ne met pas automatiquement à jour le cache sur les instances en ligne si vous modifiez ultérieurement vos livres de recettes. Si vous avez modifié vos livres de recettes depuis le début de l'instance, assurez-vous d'exécuter la commande de pile Update Cookbooks (Mettre à jour les livres de recettes) pour mettre à jour le cache des livres de recettes avec la version la plus récente du référentiel.

L'agent met en cache uniquement les commandes les plus récentes. Vous pouvez les lister en exécutant `list_commands`, qui retourne une liste des commandes mises en cache et du moment où elles ont été effectuées.

```
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
```

Pour exécuter à nouveau la commande la plus récente, exécutez ceci :

```
sudo opsworks-agent-cli run_command
```

Pour exécuter à nouveau l'instance la plus récente d'une commande spécifiée, exécutez ceci :

```
sudo opsworks-agent-cli run_command command
```

Par exemple, pour exécuter à nouveau les recettes Setup, vous pouvez exécuter la commande suivante :

```
sudo opsworks-agent-cli run_command setup
```

Chaque commande a un JSON de configuration et de déploiement de la pile associé, qui représente l'état de la pile et du déploiement au moment de l'exécution de la commande. Etant donné que ces données peuvent changer d'une commande à l'autre, une instance plus ancienne d'une commande peut utiliser des données légèrement différentes des plus récentes. Pour exécuter à nouveau une instance particulière d'une commande, copiez le temps de la sortie `list_commands` et exécutez les éléments suivants :

```
sudo opsworks-agent-cli run_command time
```

Les exemples précédents exécutent tous à nouveau la commande à l'aide du JSON par défaut, c'est-à-dire le JSON qui a été installé pour cette commande. Vous pouvez exécuter à nouveau une commande sur un fichier JSON arbitraire comme suit :

```
sudo opsworks-agent-cli run_command -f /path/to/valid/json.file
```

Affichage des journaux de Chef

La commande `show_log` de l'interface de ligne de commande de l'agent affiche un journal spécifié. Une fois la commande terminée, observez la fin du fichier. La commande `show_log` fournit donc un moyen pratique de consulter la queue du journal, où se trouvent généralement les informations relatives aux erreurs. Vous pouvez faire défiler vers le haut pour afficher les premières parties du journal.

Pour afficher le journal de la commande en cours, exécutez ceci :

```
sudo opsworks-agent-cli show_log
```

Vous pouvez également afficher les journaux d'une commande donnée, mais sachez que l'agent met en cache les journaux des trente dernières commandes seulement. Vous pouvez afficher les commandes d'une instance en exécutant `list_commands`, qui retourne une liste des commandes mises en cache et l'heure où elles ont été effectuées. Pour obtenir un exemple, consultez Exécution des recettes.

Pour afficher le journal de la dernière exécution d'une commande spécifique, exécutez ce qui suit :

```
sudo opsworks-agent-cli show_log command
```

Le paramètre de la commande peut être défini sur `setup`, `configure`, `deploy`, `undeploy`, `start`, `stop` ou `restart`. La plupart de ces commandes correspondent aux événements de cycle de vie et demandent à l'agent d'exécuter les recettes associées.

Pour afficher le journal pour une exécution de commande particulière, copiez la date de la sortie `list_commands` et exécutez :

```
sudo opsworks-agent-cli show_log date
```

Si une commande est toujours en cours d'exécution, `show_log` affiche l'état actuel du journal.

> **Note**
>
> Pour résoudre les erreurs et `show_log` les problèmes, vous out-of-memory pouvez notamment suivre un journal pendant l'exécution, comme suit :
>
> 1. Utilisez `run_command` pour déclencher l'événement du cycle de vie approprié. Pour de plus amples informations, veuillez consulter Exécution des recettes.
> 2. Exécutez plusieurs fois `show_log` pour voir la queue du journal tel qu'il est écrit.
>
> Si Chef manque de mémoire ou se ferme de façon impromptue, le journal se termine brutalement. Si une recette échoue, le journal se terminera par une exception et une trace de la pile.

Affichage du JSON de configuration et de déploiement de la pile

La plupart des données utilisées par les recettes proviennent du JSON de configuration et de déploiement de la pile, qui définit un ensemble d'attributs de Chef fournissant une description détaillée de la configuration de la pile, des déploiements et des attributs personnalisés en option que les utilisateurs peuvent ajouter. Pour chaque commande, AWS OpsWorks Stacks installe un JSON qui représente la pile et l'état du déploiement au moment de la commande. Pour de plus amples informations, veuillez consulter Attributs de déploiement et de configuration de pile.

Si vos recettes personnalisées obtiennent du JSON de configuration et de déploiement de la pile, vous pouvez vérifier les données en examinant le JSON. Le plus simple pour afficher le JSON de configuration et de déploiement de la pile consiste à exécuter la commande `get_json` de l'interface de ligne de commande de l'agent, qui affiche une version formatée de l'objet JSON. Voici les premières lignes de certaines sorties types :

```
{
  "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"
...
```

Vous pouvez afficher le JSON de configuration et de déploiement de la pile le plus récent comme suit :

```
sudo opsworks-agent-cli get_json
```

Vous pouvez afficher le JSON de configuration et de déploiement de la pile le plus récent pour une commande spécifiée en exécutant ce qui suit :

```
sudo opsworks-agent-cli get_json command
```

Le paramètre de la commande peut être défini sur `setup`, `configure`, `deploy`, `undeploy`, `start`, `stop` ou `restart`. La plupart de ces commandes correspondent aux événements de cycle de vie et demandent à l'agent d'exécuter les recettes associées.

Vous pouvez afficher le JSON de configuration et de déploiement de la pile pour une exécution de commande spécifique en spécifiant la date de la commande comme suit :

```
sudo opsworks-agent-cli get_json date
```

La façon la plus simple d'utiliser cette commande est la suivante :

1. Exécutez `list_commands`, qui retourne une liste des commandes ayant été exécutées sur l'instance et la date à laquelle chaque commande a été exécutée.
2. Copiez la date de la commande appropriée et utilisez-la comme `get_json` *`date`* argument.

Débogage et dépannage des problèmes courants

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section décrit certains problèmes de débogage et de dépannage couramment rencontrés et de leurs solutions.

Rubriques

- Résolution des problèmes liés aux AWS OpsWorks piles
- Dépannage de l'enregistrement des instances

Résolution des problèmes liés aux AWS OpsWorks piles

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Cette section présente certains problèmes courants liés à AWS OpsWorks Stacks et leurs solutions.

Rubriques

- Impossible de gérer des instances
- Après une exécution Chef, les instances ne démarrent pas
- Les instances d'une couche échouent toutes à un test de santé d'Elastic Load Balancing
- Impossible de communiquer avec un Elastic Load Balancing Load Balancer

- Une instance locale importée n'a pas réussi à finaliser la configuration du volume après un redémarrage
- Un volume EBS ne se rattache pas après un redémarrage
- Impossible de supprimer les groupes de sécurité AWS OpsWorks Stacks
- Suppression accidentelle d'un AWS OpsWorks groupe de sécurité Stacks
- Le journal Chef se termine brutalement
- Le livre de recettes n'est pas mis à jour
- Les instances sont bloquées à l'état de démarrage
- Redémarrage inattendu des instances
- opsworks-agent Processus en cours d'exécution sur les instances
- Commandes execute_recipes inattendue

Impossible de gérer des instances

Problème : Vous ne pouvez plus gérer une instance que vous pouviez gérer auparavant. Dans certains cas, les journaux peuvent afficher une erreur semblable à celle-ci :

```
Aws::CharlieInstanceService::Errors::UnrecognizedClientException - The security token
 included in the request is invalid.
```

Cause : Cela peut se produire si une ressource en dehors d' AWS OpsWorks sur laquelle dépend l'instance a été modifiée ou supprimée. Voici des exemples de modifications de ressource qui peuvent interrompre les communications avec une instance.

- Un utilisateur ou un rôle IAM associé à l'instance a été supprimé accidentellement, en dehors de AWS OpsWorks Stacks. Cela entraîne un échec de communication entre l' AWS OpsWorks agent installé sur l'instance et le service AWS OpsWorks Stacks. L'utilisateur qui est associé à une instance est requis tout au long de la durée de vie de l'instance.
- La modification de configurations de volume ou de stockage pendant qu'une instance est hors ligne peut rendre une instance ingérable.
- Ajout manuel d' EC2 instances à un ELB. AWS OpsWorks reconfigure un équilibreur de charge Elastic Load Balancing attribué chaque fois qu'une instance entre ou quitte l'état en ligne. AWS OpsWorks considère uniquement les instances dont il sait qu'elles sont des membres valides ; les instances ajoutées en dehors de AWS OpsWorks ou par un autre processus sont supprimées. Toute instance est supprimée.

Solution : ne supprimez pas les utilisateurs ou les rôles IAM dont dépendent vos instances. Si possible, modifiez des configurations de volume ou de stockage uniquement tant que les instances dépendantes s'exécutent. AWS OpsWorks À utiliser pour gérer l'équilibreur de charge ou les adhésions EIP des instances. AWS OpsWorks Lorsque vous enregistrez une instance, pour éviter les problèmes liés à la gestion des instances enregistrées en cas de suppression accidentelle de l'utilisateur, ajoutez le `--use-instance-profile` paramètre à votre `register` commande pour utiliser le profil d'instance intégré de l'instance à la place.

Après une exécution Chef, les instances ne démarrent pas

Problème : sur les piles Chef 11.10 ou antérieures qui sont configurées pour utiliser les livres de recettes personnalisés, après qu'une exécution Chef a utilisée les livres de recettes de la Communauté, les instances ne démarrent pas. Les messages des journaux peuvent établir que les recettes n'ont pas pu être compilées (« Erreur de compilation de recette ») ou n'ont pas pu être chargées, car elles ne peuvent pas trouver une dépendance.

Cause : la cause la plus probable est que le livre de recettes personnalisé ou le livre de recettes de la communauté ne prend pas en charge la version Chef qu'utilise votre pile. Certains livres populaires de la communauté, tels que `apt` et `build-essential`, ont connu des problèmes de compatibilité avec Chef 11.10.

Solution : Sur les AWS OpsWorks piles dont le paramètre Utiliser les livres de recettes Chef personnalisés est activé, les livres de recettes personnalisés ou communautaires doivent toujours prendre en charge la version de Chef utilisée par votre pile. Epinglez les livres de recettes de la communauté avec une version (à savoir, définissez le numéro de version du livre de recettes sur une version spécifique) qui est compatible avec la version Chef configurée dans les paramètres de votre pile. Pour trouver une version du livre de recettes de la communauté prise en charge, affichez le journal des modifications d'un livre des recettes dont la compilation a échoué et utilisez uniquement la version la plus récente du livre de recettes que votre pile prendra en charge. Pour épingler une version du livre de recettes, spécifiez un numéro de version exacte dans le fichier Berksfile du référentiel de votre livre de recettes personnalisé. Par exemple, `cookbook 'build-essential', '= 3.2.0'`.

Les instances d'une couche échouent toutes à un test de santé d'Elastic Load Balancing

Problème : vous attachez un équilibreur de charge Elastic Load Balancing à une couche de serveur d'applications, mais le bilan de santé de toutes les instances échoue.

Cause : Lorsque vous créez un équilibreur de charge Elastic Load Balancing, vous devez spécifier le chemin ping que l'équilibreur de charge appelle pour déterminer si l'instance est saine. Assurez-

vous de spécifier un chemin d'accès ping approprié pour votre application ; la valeur par défaut est/index.html. Si votre application n'inclut pas un `index.html`, vous devez spécifier un chemin d'accès approprié ou la vérification du statut échoue. Par exemple, l'PHPApp application Simple utilisée dans Mise en route des piles Linux Chef 11 n'utilise pas index.html ; le chemin ping approprié pour ces serveurs est /.

Solution : Modifiez le chemin d'accès ping de l'équilibreur de charge. Pour plus d'informations, consultez Elastic Load Balancing.

Impossible de communiquer avec un Elastic Load Balancing Load Balancer

Problème : vous créez un équilibreur de charge Elastic Load Balancing et vous l'associez à une couche de serveur d'applications, mais lorsque vous cliquez sur le nom DNS ou l'adresse IP de l'équilibreur de charge pour exécuter l'application, le message d'erreur suivant s'affiche : « Le serveur distant ne répond pas ».

Cause : Si votre stack s'exécute dans un VPC par défaut, lorsque vous créez un équilibreur de charge Elastic Load Balancing dans la région, vous devez spécifier un groupe de sécurité. Le groupe de sécurité doit avoir les règles de trafic entrant qui autorisent le trafic entrant à partir de votre adresse IP. Si vous spécifiez default VPC security group (groupe de sécurité VPC par défau), la règle de trafic entrant par défaut n'accepte aucun trafic entrant.

Solution : Modifiez les règles de trafic entrant du groupe de sécurité pour accepter le trafic entrant à partir des adresses IP appropriées.

1. Cliquez sur Security Groups dans le volet de navigation de la EC2 console Amazon.
2. Sélectionnez le groupe de sécurité de l'équilibreur de charge.
3. Cliquez sur Modifier sous l'onglet Entrant.
4. Ajoutez une règle de trafic entrant avec le champ Source défini avec un CIDR approprié.

   Par exemple, la spécification Anywhere définit le CIDR avec la valeur 0.0.0.0/0, qui demande à l'équilibreur de charge d'accepter le trafic entrant à partir de n'importe quelle adresse IP.

Une instance locale importée n'a pas réussi à finaliser la configuration du volume après un redémarrage

Problème : vous redémarrez une EC2 instance que vous avez importée dans AWS OpsWorks Stacks, et la console AWS OpsWorks Stacks affiche le statut de l'instance « échec ». Cela peut se produire sur les instances Chef 11 ou Chef 12.

Cause : AWS OpsWorks Stacks ne parvient pas à attacher un volume à votre instance pendant le processus d'installation. Une cause possible est qu' AWS OpsWorks Stacks remplace votre configuration de volume sur votre instance lorsque vous exécutez la commande `setup`.

Solution : ouvrez la page Détails de l'instance et vérifiez votre configuration de volume dans la zone Volumes. Notez que vous pouvez changer la configuration de votre volume uniquement lorsque votre instance se trouve dans l'état stopped (arrêté). N'oubliez pas que chaque volume a un nom et un point de montage spécifiés. Vérifiez que vous avez fourni le bon point de montage dans votre configuration dans AWS OpsWorks Stacks avant de redémarrer l'instance.

Un volume EBS ne se rattache pas après un redémarrage

Problème : vous utilisez la EC2 console Amazon pour associer un volume Amazon EBS à une instance, mais lorsque vous redémarrez l'instance, le volume n'est plus attaché.

Cause : AWS OpsWorks Stacks ne peut rattacher que les volumes Amazon EBS dont il a connaissance, qui sont limités aux volumes suivants :

- Volumes créés par AWS OpsWorks Stacks.
- Volumes de votre compte que vous avez explicitement enregistrés auprès d'une pile à l'aide de la page Ressources.

Solution : Gérez vos volumes Amazon EBS uniquement à l'aide de la console, de l'API ou de la CLI AWS OpsWorks Stacks. Si vous souhaitez utiliser l'un des volumes Amazon EBS de votre compte avec une pile, utilisez la page Ressources de la pile pour enregistrer le volume et l'associer à une instance. Pour de plus amples informations, veuillez consulter Gestion des ressources.

Impossible de supprimer les groupes de sécurité AWS OpsWorks Stacks

Problème : une fois que vous avez supprimé une pile, il reste un certain nombre de groupes de sécurité AWS OpsWorks Stacks qui ne peuvent pas être supprimés.

Cause : Les groupes de sécurité doivent être supprimés selon un ordre particulier.

Solution : Tout d'abord, assurez-vous qu'aucune instance n'utilise les groupes de sécurité. Puis, supprimez l'un des groupes de sécurité ci-après, s'ils existent, dans l'ordre suivant :

1. AWS- OpsWorks -Blank-Server
2. AWS- OpsWorks -Monitoring-Master-Server
3. Serveur AWS- OpsWorks -DB-Master

4. Serveur AWS- OpsWorks -Memcached-Server
5. AWS- OpsWorks -Serveur personnalisé
6. Serveur d'applications AWS- OpsWorks -NodeJS
7. Serveur d'applications AWS- OpsWorks -PHP
8. Serveur d'applications AWS- OpsWorks -Rails
9. AWS- OpsWorks -Serveur Web
10. AWS- OpsWorks -Serveur par défaut
11. Serveur AWS- OpsWorks -LB

Suppression accidentelle d'un AWS OpsWorks groupe de sécurité Stacks

Problème : vous avez supprimé l'un des groupes de sécurité AWS OpsWorks Stacks et vous devez le recréer.

Cause : Ces groupes de sécurité sont généralement supprimés par accident.

Solution : Le groupe recréé doit être une copie exacte de l'original, y compris la même casse pour le nom du groupe. Au lieu de recréer le groupe manuellement, l'approche par défaut consiste à ce qu' AWS OpsWorks Stacks exécute la tâche automatiquement. Créez simplement une nouvelle pile dans la même région AWS (et un VPC, le cas échéant AWS OpsWorks). Stacks recréera automatiquement tous les groupes de sécurité intégrés, y compris celui que vous avez supprimé. Vous pouvez ensuite supprimer la pile si vous n'en avez plus l'utilisation ; les groupes de sécurité demeureront.

Le journal Chef se termine brutalement

Problème : Un journal Chef se termine brutalement ; la fin du journal n'indique pas une exécution réussie ni n'affiche une exception et une trace de pile.

Cause : Ce comportement est généralement dû à une mémoire insuffisante.

Solution : Créez une plus grande instance et utilisez la commande `run_command` de l'interface de ligne de commande de l'agent pour exécuter les recettes à nouveau. Pour de plus amples informations, veuillez consulter Exécution des recettes.

Le livre de recettes n'est pas mis à jour

Problème : Vous avez mis à jour vos livres de recettes, mais les instances de la pile exécutent toujours les anciennes recettes.

Cause : AWS OpsWorks Stacks met en cache les livres de recettes sur chaque instance et exécute les recettes à partir du cache, et non du référentiel. Lorsque vous démarrez une nouvelle instance, AWS OpsWorks Stacks télécharge vos livres de recettes du référentiel vers le cache de l'instance. Cependant, si vous modifiez par la suite vos livres de recettes personnalisés, AWS OpsWorks Stacks ne met pas automatiquement à jour les caches des instances en ligne.

Solution : exécutez la commande Update Cookbooks stack pour demander explicitement à AWS OpsWorks Stacks de mettre à jour les caches de livres de recettes de vos instances en ligne.

Les instances sont bloquées à l'état de démarrage

Problème : lorsque vous redémarrez une instance ou que la réparation automatique la redémarre, l'opération de démarrage se bloque sur l'état `booting`.

Cause : une cause possible de ce problème est la configuration du VPC, VPC par défaut inclus. Les instances doivent toujours être en mesure de communiquer avec le service AWS OpsWorks Stacks, Amazon S3, ainsi qu'avec les référentiels de packages, de livres de recettes et d'applications. Si, par exemple, vous supprimez une passerelle par défaut d'un VPC par défaut, les instances perdent leur connexion au service AWS OpsWorks Stacks. Comme AWS OpsWorks Stacks ne peut plus communiquer avec l'agent d'instance, il considère l'instance comme défaillante et la répare automatiquement. Cependant, sans connexion, AWS OpsWorks Stacks ne peut pas installer d'agent d'instance sur l'instance réparée. Sans agent, AWS OpsWorks Stacks ne peut pas exécuter les recettes de configuration sur l'instance, de sorte que l'opération de démarrage ne peut pas progresser au-delà de l'état de « démarrage ».

Solution : Modifiez la configuration de votre VPC afin que les instances aient la connectivité requise.

Redémarrage inattendu des instances

Problème : une instance arrêtée redémarre de façon inattendue.

Cause 1 : si vous avez activé la réparation automatique pour vos instances, AWS OpsWorks Stacks vérifie régulièrement l'état des EC2 instances Amazon associées et redémarre celles qui ne fonctionnent pas correctement. Si vous arrêtez ou mettez fin à une instance AWS OpsWorks gérée par Stacks à l'aide de la EC2 console, de l'API ou de la CLI Amazon, AWS OpsWorks Stacks n'en est pas informé. Au lieu de cela, il considère l'instance arrêtée comme défectueuse et la redémarre automatiquement.

Solution : Gérez vos instances uniquement avec la console AWS OpsWorks Stacks, l'API ou l'interface de ligne de commande. Si vous utilisez AWS OpsWorks Stacks pour arrêter ou supprimer

une instance, elle ne sera pas redémarrée. Pour plus d'informations, consultez Démarrage, arrêt et redémarrage manuels des instances 24/7 et Supprimer des AWS OpsWorks instances de Stacks.

Cause 2 : Les instances peuvent échouer pour un grand nombre de raisons. Si la guérison automatique est activée, AWS OpsWorks Stacks redémarre automatiquement les instances défaillantes.

Solution : Il s'agit d'un fonctionnement normal ; il n'est pas nécessaire de faire quoi que ce soit sauf si vous ne voulez pas que AWS OpsWorks Stacks redémarre les instances défaillantes. Dans ce cas, vous devez désactiver la réparation automatique.

`opsworks-agent` Processus en cours d'exécution sur les instances

Problème : Plusieurs processus `opsworks-agent` sont en cours d'exécution sur vos instances. Par exemple :

```
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
```

Cause : il s'agit de processus légitimes qui sont obligatoires pour le fonctionnement normal de l'agent. Ils exécutent des tâches telles que la gestion des déploiements et l'envoi de messages de connexion persistante au service.

Solution : il s'agit du comportement normal. N'arrêtez pas ces processus, sinon vous compromettrez le fonctionnement de l'agent.

Commandes execute_recipes inattendue

Problème : la section Logs de la page des détails d'une instance inclut des commandes `execute_recipes` inattendues. Des commandes `execute_recipes` inattendues peuvent également apparaître sur les pages Stack et Deployments.

Cause : ce problème est souvent provoqué par des modifications d'autorisation. Lorsque vous modifiez les autorisations SSH ou sudo d'un utilisateur ou d'un groupe, AWS OpsWorks Stacks

s'exécute `execute_recipes` pour mettre à jour les instances de la pile et déclenche également un événement Configure. Une autre source de commandes `execute_recipes` inattendues est la mise à jour par AWS OpsWorks Stacks de l'agent de l'instance.

Solution : Il s'agit d'un fonctionnement normal ; nul besoin de faire quoi que ce soit.

Pour voir quelles actions une commande `execute_recipes` exécute, accédez à la page Déploiements et cliquez sur l'horodatage de la commande. La page des détails de la commande s'ouvre et affiche les principales recettes qui ont été exécutées. Par exemple, la page de détails suivante concerne une commande `execute_recipes` qui a lancé `ssh_users` pour mettre à jour les autorisations SSH.



Pour voir tous les détails, cliquez sur afficher dans la colonne Journal de la commande afin d'afficher le journal Chef associé. Recherchez dans le journal**`Run List`**. AWS OpsWorks Les recettes d'entretien des piles seront présentées ci-dessous. `OpsWorks Custom Run List` Par exemple, la liste suivante est la liste des exécutions de la commande `execute_recipes` affichée dans la capture d'écran précédente et illustre chaque recette associée à la commande.

```
[2014-02-21T17:16:30+00:00] INFO: OpsWorks Custom Run List:
 ["opsworks_stack_state_sync",
  "ssh_users", "test_suite", "opsworks_cleanup"]
```

Dépannage de l'enregistrement des instances

Cette section contient quelques problèmes d'enregistrement des instances couramment rencontrés et leurs solutions.

> **Note**
>
> Si vous rencontrez des problèmes d'enregistrement, exécutez `register` avec l'argument `--debug`, qui fournit des informations de débogage supplémentaires.

Rubriques

- EC2L'utilisateur n'est pas autorisé à effectuer :...
- Les informations d'identification doivent être limitées à une région valide

EC2L'utilisateur n'est pas autorisé à effectuer :...

Problème : Une commande `register` renvoie quelque chose de similaire à ce qui suit :

```
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
```

Cause : La `register` commande est exécutée avec des informations d'identification qui n'accordent pas les autorisations requises. La stratégie de l'utilisateur doit permettre l'action `iam:CreateGroup`, entre autres.

Solution : Fournissez à `register` les informations d'identification utilisateur IAM qui disposent des autorisations requises. Pour de plus amples informations, veuillez consulter Installation et configuration de l' AWS CLI.

Les informations d'identification doivent être limitées à une région valide

Problème : Une commande `register` renvoie les éléments suivants :

```
A client error (InvalidSignatureException) occurred when calling the
DescribeStacks operation: Credential should be scoped to a valid region, not 'cn-
north-1'.
```

Cause : La région de la commande doit être une région AWS OpsWorks Stacks. Pour obtenir une liste des régions prises en charge, consultez la page Prise en charge de la région. En général, cette erreur se produit pour l'une des raisons suivantes :

- La pile est dans une autre région et vous avez attribué une région de la pile à l'argument `--region` de la commande.

  Il n'est pas nécessaire de spécifier une région de pile ; AWS OpsWorks Stacks la détermine automatiquement à partir de l'ID de pile.
- Vous avez omis l'argument `--region` qui spécifie implicitement la région par défaut, mais votre région par défaut n'est pas prise en charge par AWS OpsWorks Stacks.

Solution : définissez `--region` explicitement une région AWS OpsWorks Stacks prise en charge, ou modifiez votre AWS CLI `config` fichier pour remplacer la région par défaut par une région AWS OpsWorks Stacks prise en charge. Pour de plus amples informations, veuillez consulter Configuration de l'interface de ligne de commande AWS.

AWS OpsWorks CLI de l'agent Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

> **Note**
>
> Cette fonctionnalité n'est disponible que sur les instances Linux.

L'agent que AWS OpsWorks Stacks installe sur chaque instance expose une interface de ligne de commande (CLI). Si vous utilisez SSH pour vous connecter à l'instance, vous pouvez utiliser l'interface de ligne de commande pour les actions suivantes :

- Accéder aux fichiers journaux pour les exécutions Chef.
- Commandes Access AWS OpsWorks Stacks.
- Exécuter manuellement les recettes Chef.

- Afficher les rapports de l'instance.
- Afficher les rapports de l'agent.
- Afficher un ensemble limité d'attributs de déploiement et de configuration de la pile.

> **Important**
>
> Vous pouvez exécuter les commandes de l'interface de ligne de commande de l'agent en tant que racine ou à l'aide de `sudo`.

La syntaxe de base de la commande est :

```
sudo opsworks-agent-cli [--help] [command [activity] [date]]
```

Les quatre arguments sont les suivants :

help

(Facultatif) Affiche un bref résumé des commandes disponibles, lorsqu'il est utilisé seul. Lorsqu'il est utilisé avec une commande, `help` affiche une description de la commande.

commande

(Facultatif) Commande de l'interface de ligne de commande de l'agent, qui doit être définie sur l'une des valeurs suivantes :

- agent_report
- get_json
- instance_report
- list_commands
- run_command
- show_log
- stack_state

activity

(Facultatif) Utilisé comme argument avec certaines commandes pour spécifier une activité AWS OpsWorks Stacks particulière : `setup`, `configure`, `deploy`, `undeploy`, `start`, `stop` ou `restart`.

date

(Facultatif) Utilisé comme argument avec certaines commandes pour spécifier une exécution de commande AWS OpsWorks Stacks particulière. Spécifiez l'exécution de la commande en définissant la date à laquelle la commande a été exécutée au *yyyy-mm-ddThh:mm:ss* format, y compris les guillemets simples. Par exemple, pour 10:31:55 le mardi 5 février 2013, utilisez : `'2013-02-05T10:31:55'`. Pour déterminer à quel moment une commande AWS OpsWorks Stacks particulière a été exécutée, exécutez list_commands.

> **Note**
>
> Si l'agent a exécuté plusieurs fois la même activité AWS OpsWorks Stacks, vous pouvez sélectionner une exécution particulière en spécifiant à la fois l'activité et l'heure à laquelle elle a été exécutée. Si vous spécifiez une activité et omettez l'heure, la commande de l'interface de ligne de commande de l'agent agit sur la plus récente exécution de l'activité. Si vous omettez les deux arguments, la commande de l'interface de ligne de commande de l'agent intervient sur l'activité la plus récente.

Les sections suivantes décrivent les commandes et leurs arguments associés. Pour des raisons de concision, les sections de syntaxe omettent l'option `--help`, qui peut être utilisée avec n'importe quelle commande.

Rubriques

- agent_report
- get_json
- instance_report
- list_commands
- run_command
- show_log
- stack_state

agent_report

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Retourne un rapport de l'agent.

```
sudo opsworks-agent-cli agent_report
```

L'exemple de sortie suivant correspond à une instance qui a très récemment lancé une activité de configuration.

```
$ 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**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Retourne des informations sur une exécution Chef en tant qu'objet JSON.

```
sudo opsworks-agent-cli get_json [activity] [date] [-i | --internal | --no-i | --no-
internal]
```

Par défaut, `get_json` affiche les informations fournies par les clients pour l'exécution Chef la plus récente. Utilisez les options suivantes pour spécifier un ensemble particulier d'informations.

activity

Affiche les informations pour l'exécution Chef associée à l'activité spécifiée la plus récente. Pour obtenir une liste des activités valides, exécutez `list_commands`.

date

Affiche les informations sur l'exécution Chef associée à l'activité exécutée pour l'horodatage spécifié. Pour obtenir la liste des horodatages valides, exécutez `list_commands`.

-i, --internal

Affiche les informations que AWS OpsWorks Stacks utilise en interne pour l'exécution de Chef.

--no-i, --no-internal

Affiche explicitement les informations fournies par les clients pour l'exécution Chef. Valeur par défaut, sauf spécification contraire.

> **Note**
>
> Pour les instances Linux de Chef 12, l'exécution de cette commande renvoie les informations valides, telles que les attributs de déploiement et de configuration de pile de l'instance. Toutefois, pour obtenir des informations plus complètes, reportez-vous aux sacs de données Chef créés par AWS OpsWorks Stacks sur l'instance. Pour plus d'informations, consultez le AWS OpsWorks Référence du sac de données Stacks.

L'exemple de sortie suivant montre les informations fournies par les clients pour l'exécution Chef la plus récente de l'activité de configuration la plus récente.

```
$ sudo opsworks-agent-cli get_json configure

{
  "run_list": [
```

```
    "recipe[opsworks_cookbook_demo::configure]"
  ]
}
```

L'exemple de sortie suivant montre les informations que AWS OpsWorks Stacks utilise en interne pour l'exécution de Chef exécutée pour l'horodatage spécifié.

```
$ sudo opsworks-agent-cli get_json 2015-12-01T18:20:24 -i

{
  "aws_opsworks_agent": {
    "version": "4004-20151201152533",
    "valid_client_activities": [
      "reboot",
      "stop",
      "deploy",
      "grant_remote_access",
      "revoke_remote_access",
      "update_agent",
      "setup",
      "configure",
      "update_dependencies",
      "install_dependencies",
      "update_custom_cookbooks",
      "execute_recipes",
      "sync_remote_users"
    ],
    "command": {
      "type": "configure",
      "args": {
        "app_ids": [

        ]
      },
      "sent_at": "2015-12-01T18:19:23+00:00",
      "command_id": "5c2113f3-c6d5-40eb-bcfa-77da2885eeEX",
      "iam_user_arn": null,
      "instance_id": "cfdaa716-42fe-4e3b-9762-fef184ddd8EX"
    },
    "resources": {
      "apps": [

      ],
```

```
      "layers": [
        {
          "layer_id": "93f50d83-1e73-45c4-840a-0d4f07cda1EX",
          "name": "MyCookbooksDemoLayer",
          "packages": [

          ],
          "shortname": "cookbooks-demo",
          "type": "custom",
          "volume_configurations": [

          ]
        }
      ],
      "instances": [
        {
          "ami_id": "ami-d93622EX",
          "architecture": "x86_64",
          "auto_scaling_type": null,
          "availability_zone": "us-west-2a",
          "created_at": "2015-11-18T00:21:05+00:00",
          "ebs_optimized": false,
          "ec2_instance_id": "i-a480e960",
          "elastic_ip": null,
          "hostname": "cookbooks-demo1",
          "instance_id": "cfdaa716-42fe-4e3b-9762-fef184ddd8EX",
          "instance_type": "c3.large",
          "layer_ids": [
            "93f50d83-1e73-45c4-840a-0d4f07cda1EX"
          ],
          "os": "Amazon Linux 2015.09",
          "private_dns": "ip-192-0-2-0.us-west-2.compute.internal",
          "private_ip": "10.122.69.33",
          "public_dns": "ec2-203-0-113-0.us-west-2.compute.amazonaws.com",
          "public_ip": "192.0.2.0",
          "root_device_type": "ebs",
          "root_device_volume_id": "vol-f6f7e8EX",
          "ssh_host_dsa_key_fingerprint": "f2:...:15",
          "ssh_host_dsa_key_public": "ssh-dss AAAAB3Nz...a8vMbqA=",
          "ssh_host_rsa_key_fingerprint": "0a:...:96",
          "ssh_host_rsa_key_public": "ssh-rsa AAAAB3Nz...yhPanvo7",
          "status": "online",
          "subnet_id": null,
          "virtualization_type": "paravirtual",
```

```
          "infrastructure_class": "ec2",
          "ssh_host_dsa_key_private": "-----BEGIN DSA PRIVATE KEY-----
\nMIIDVwIB...g5OtgQ==\n-----END DSA PRIVATE KEY-----\n",
          "ssh_host_rsa_key_private": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIB...78kprtIw\n-----END RSA PRIVATE KEY-----\n"
        }
      ],
      "users": [

      ],
      "elastic_load_balancers": [

      ],
      "rds_db_instances": [

      ],
      "stack": {
        "arn": "arn:aws:opsworks:us-west-2:80398EXAMPLE:stack/040c3def-b2b4-4489-bb1b-
e08425886fEX/",
        "custom_cookbooks_source": {
          "type": "s3",
          "url": "https://s3.amazonaws.com/amzn-s3-demo-bucket/opsworks-cookbook-
demo.tar.gz",
          "username": "AKIAJUQN...WG644EXA",
          "password": "O5v+4Zz+...rcKbFTJu",
          "ssh_key": null,
          "revision": null
        },
        "name": "MyCookbooksDemoStack",
        "region": "us-west-2",
        "stack_id": "040c3def-b2b4-4489-bb1b-e08425886fEX",
        "use_custom_cookbooks": true,
        "vpc_id": null
      },
      "ecs_clusters": [

      ],
      "volumes": [

      ]
    },
    "chef": {
      "customer_recipes": [
        "opsworks_cookbook_demo::configure"
```

```
      ],
      "customer_json": "e30=\n",
      "customer_data_bags": "e30=\n"
    }
  }
}
```

instance_report

> **Important**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Retourne un rapport d'instance étendu.

```
sudo opsworks-agent-cli instance_report
```

L'exemple de sortie suivant provient d'une 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**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Affiche les heures auxquelles chaque activité a été exécutée sur cette instance. Vous pouvez utiliser ces heures pour que d'autres commandes de l'interface de ligne de commande de l'agent spécifient une exécution particulière.

```
sudo opsworks-agent-cli list_commands  [activity] [date]
```

L'exemple de sortie suivant correspond à une instance qui a lancé des activités de configuration, de paramétrage et de mise à jour des livres de recettes personnalisés.

```
$ 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**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Exécute une commande AWS OpsWorks Stacks, qui est un fichier JSON contenant une liste d'exécution Chef contenant les informations nécessaires pour exécuter une activité AWS OpsWorks Stacks (installation, configuration, déploiement, etc.). La commande `run_command` génère une entrée de journal que vous pouvez afficher en exécutant `show_log`. Cette option est uniquement destinée à des fins de développement. AWS OpsWorks Stacks ne suit donc pas les modifications.

```
sudo opsworks-agent-cli run_command [activity] [date] [/path/to/valid/json.file]
```

`run_command`Exécute par défaut la commande AWS OpsWorks Stacks la plus récente. Utilisez les options suivantes pour spécifier une commande particulière.

activity

Exécutez une commande AWS OpsWorks Stacks spécifiée :`setup`,`configure`,`deploy`,`undeploy`, `startstop`, our`estart`.

date

Exécutez la OpsWorks commande AWS exécutée à l'horodatage spécifié. Pour obtenir la liste des horodatages valides, exécutez `list_commands`.

file

Exécutez le fichier JSON de commande spécifié. Pour obtenir le chemin d'accès d'une commande, exécutez `get_json`.

L'exemple de sortie suivant provient d'une instance et exécute la commande « configure ».

```
$ 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**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Retourne le fichier journal d'une commande.

```
sudo opsworks-agent-cli show_log [activity] [date]
```

Par défaut, `show_log` suit le fichier journal le plus récent. Utilisez les options suivantes pour spécifier une commande particulière.

activity

Afficher le fichier journal de l'activité spécifiée.

date

Afficher le fichier journal de l'activité exécutée à l'horodatage spécifié. Pour obtenir la liste des horodatages valides, exécutez `list_commands`.

L'exemple de sortie suivant indique les journaux les plus récents.

```
$ 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**
>
> Le AWS OpsWorks Stacks service a pris fin le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Affiche les informations que AWS OpsWorks Stacks utilise en interne pour la dernière exécution de Chef.

```
opsworks-agent-cli stack_state
```

> **Note**
>
> Pour les instances Linux de Chef 12, l'exécution de cette commande renvoie les informations valides, telles que les attributs de déploiement et de configuration de pile de l'instance. Toutefois, pour obtenir des informations plus complètes, reportez-vous aux sacs de données Chef créés par AWS OpsWorks Stacks sur l'instance. Pour plus d'informations, consultez le AWS OpsWorks Référence du sac de données Stacks.

L'exemple de sortie suivant provient d'une 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/amzn-s3-demo-bucket/opsworks-cookbook-
demo.tar.gz",
      "username": "AKIAJUQN...WG644EXA",
      "password": "O5v+4Zz+...rcKbFTJu",
      "ssh_key": null,
      "revision": null
    },
    "name": "MyCookbooksDemoStack",
    "region": "us-west-2",
    "stack_id": "040c3def-b2b4-4489-bb1b-e08425886fEX",
    "use_custom_cookbooks": true,
    "vpc_id": null
  },
  "agent": {
    "valid_activities": [
      "reboot",
      "stop",
      "deploy",
      "grant_remote_access",
      "revoke_remote_access",
      "update_agent",
      "setup",
      "configure",
      "update_dependencies",
      "install_dependencies",
      "update_custom_cookbooks",
      "execute_recipes",
      "sync_remote_users"
    ]
  }
}
```

AWS OpsWorks Référence du sac de données Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

AWS OpsWorks Stacks expose une grande variété de paramètres aux recettes sous forme de contenu de sac de données Chef. Cette référence répertorie le contenu des conteneurs de données.

Un conteneur de données est un concept Chef. Un conteneur de données est une variable globale qui est stockée en tant que données JSON sur une instance ; les données JSON sont accessibles à partir de Chef. Par exemple, un sac de données peut stocker des variables globales telles que l'URL source d'une application, le nom d'hôte de l'instance et l'identifiant VPC de la pile associée. AWS OpsWorks Stacks stocke ses sacs de données sur les instances de chaque pile. Sur les instances Linux, AWS OpsWorks Stacks stocke les sacs de données dans le `/var/chef/runs/`*`run-ID`*`/data_bags` répertoire. Sur les instances Windows, il stocke les conteneurs de données dans le répertoire *`drive`*`:\chef\runs\`*`run-id`*`\data_bags`. Dans les deux cas, *`run-ID`* il s'agit d'un identifiant unique que AWS OpsWorks Stacks attribue à chaque exécution de Chef sur une instance. Ces répertoires incluent un ensemble de conteneurs de données (sous-répertoires). Chaque conteneur de données contient zéro ou plusieurs éléments de conteneurs de données, fichiers au format JSON contenant les ensembles des contenus des conteneurs de données.

> **Note**
>
> AWS OpsWorks Stacks ne prend pas en charge les sacs de données cryptés. Pour stocker des données sensibles sous une forme chiffrée, telle que mots de passe ou certificats, nous vous recommandons de les stocker dans un compartiment S3 privé. Vous pouvez ensuite créer une recette personnalisée qui utilise le Kit SDK Amazon pour Ruby pour récupérer les données. Pour obtenir un exemple, consultez Utilisation du kit SDK pour Ruby.

Le contenu d'un conteneur de données peut inclure les éléments suivants :

- Contenu chaîne qui respecte la syntaxe Ruby standard et peut utiliser des guillemets simples ou doubles, même si les chaînes contenant des caractères spéciaux doivent avoir des guillemets doubles. Pour plus d'informations, consultez le site de la documentation Ruby.
- Contenu booléen, qui est soit `true` soit `false` (sans guillemets).
- Contenu numérique, qui correspond à des nombres entiers ou décimaux, tels que `4` ou `2.5` (sans guillemets).
- Contenu liste, qui prend la forme de valeurs séparées par des virgules entre crochets (sans guillemets), comme `[ '80', '443' ]`
- Objets JSON, qui comportent un contenu supplémentaire de conteneur de données, comme `"my-app": {"elastic_ip": null,...}`.

Les recettes Chef peuvent accéder aux conteneurs de données, aux éléments des conteneurs de données et au contenu des conteneurs de données via la recherche Chef ou directement. Ce qui suit explique comment utiliser les deux approches d'accès (même si la recherche Chef est privilégiée).

Pour accéder à un sac de données via Chef Search, utilisez la méthode de recherche en spécifiant l'index de recherche souhaité. AWS OpsWorks Stacks fournit les index de recherche suivants :

- aws_opsworks_app, qui représente un ensemble d'applications déployées pour une pile.
- aws_opsworks_command, qui représente un ensemble de commandes exécutées sur une pile.
- aws_opsworks_ecs_cluster, qui représente un ensemble d'instances de cluster Amazon Elastic Container Service (Amazon ECS) pour une pile.
- aws_opsworks_elastic_load_balancer, qui représente un ensemble d'équilibreurs de charge Elastic Load Balancing pour une pile.
- aws_opsworks_instance, qui représente un ensemble d'instances d'une pile.
- aws_opsworks_layer, qui représente un ensemble de couches d'une pile.
- aws_opsworks_rds_db_instance, qui représente un ensemble d'instances Amazon Relational Database Service (Amazon RDS) pour une pile.
- aws_opsworks_stack, qui représente une pile.
- aws_opsworks_user, qui représente un ensemble d'utilisateurs d'une pile.

Une fois que vous connaissez le nom de l'index de recherche, vous pouvez accéder au contenu du sac données de cet index de recherche. Par exemple, le code de recette suivant utilise l'index de recherche `aws_opsworks_app` pour obtenir le contenu du premier élément du conteneur de

données (le premier fichier JSON) du conteneur de données `aws_opsworks_app` (le répertoire `aws_opsworks_app`). Le code écrit ensuite deux messages dans le journal Chef, l'un avec le contenu du conteneur de données du nom court de l'application (une chaîne du fichier JSON) et l'autre avec le contenu du conteneur de données de l'URL source de l'application (une autre chaîne du fichier JSON) :

```
app = search("aws_opsworks_app").first
Chef::Log.info("********** The app's short name is '#{app['shortname']}' **********")
Chef::Log.info("********** The app's URL is '#{app['app_source']['url']}' **********")
```

Où `['shortname']` et `['app_source']['url']` spécifient le contenu du conteneur de données suivant du fichier JSON correspondant :

```
{
  ...
  "shortname": "mylinuxdemoapp",
  ...
  "app_source": {
    ...
    "url": "https://s3.amazonaws.com/opsworks-demo-assets/opsworks-linux-demo-
nodejs.tar.gz",
  },
  ...
}
```

Pour obtenir la liste du contenu de conteneur de données que vous pouvez rechercher, consultez les rubriques de référence de cette section.

Vous pouvez également parcourir un ensemble d'éléments d'un conteneur de données. Par exemple, le code de recette suivant est similaire à l'exemple précédent ; il parcourt chacun des éléments du conteneur de données lorsqu'il y a plusieurs éléments :

```
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
```

Si vous savez qu'il existe un contenu de conteneur de données spécifique, vous pouvez trouver l'élément de conteneur de données correspondant avec la syntaxe suivante :

```
search("search_index", "key:value").first
```

Par exemple, le code de recette suivant utilise l'index de recherche `aws_opsworks_app` pour trouver l'élément de conteneur de données qui contient le nom court de `mylinuxdemoapp`. Il utilise ensuite le contenu de l'élément de conteneur de données pour écrire un message dans le journal Chef avec le nom court et l'URL source de l'application correspondante :

```
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']}' **********")
```

Pour l'index de recherche `aws_opsworks_instance` uniquement, vous pouvez spécifier `self:true` pour représenter l'instance sur laquelle la recette est exécutée. Le code de recette suivant utilise le contenu de l'élément du sac de données correspondant pour écrire un message dans le journal Chef avec l'ID et le système d'exploitation AWS OpsWorks générés par Stacks de l'instance correspondante :

```
instance = search("aws_opsworks_instance", "self:true").first
Chef::Log.info("********** For instance '#{instance['instance_id']}', the instance's
 operating system is '#{instance['os']}' **********")
```

Au lieu d'utiliser la recherche Chef pour accéder aux conteneurs de données, aux éléments des conteneurs de données et au contenu des conteneurs de données, vous pouvez y accéder directement. Pour ce faire, utilisez les méthodes data_bag et data_bag_item pour accéder respectivement aux conteneurs de données et aux éléments des conteneurs de données. Par exemple, le code de recette suivant fait les mêmes choses que les exemples précédents, sauf qu'il accède directement à un seul élément de conteneur de données, puis à plusieurs éléments de conteneur de données lorsqu'il y en a plusieurs :

```
# 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
```

Entre ces deux approches, nous vous recommandons d'utiliser la recherche Chef. Tous les exemples associés de ce guide illustrent cette approche.

Rubriques

- Conteneur de données d'application (aws_opsworks_app)
- Conteneurs de données de commandes (aws_opsworks_command)
- Sac de données du cluster Amazon ECS (aws_opsworks_ecs_cluster)
- Sac de données Elastic Load Balancing (aws_opsworks_elastic_load_balancer)
- Conteneur de données de l'instance (aws_opsworks_instance)
- Conteneur de données de la couche (aws_opsworks_layer)
- Sac de données Amazon RDS (aws_opsworks_rds_db_instance)
- Conteneur de données de la pile (aws_opsworks_stack)
- Conteneur de données utilisateur (aws_opsworks_user)

Conteneur de données d'application (aws_opsworks_app)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Pour un événement Deploy ou une commande de pile Execute Recipes, représente les paramètres d'une application.

L'exemple suivant montre comment utiliser la recherche Chef pour rechercher dans un seul élément du sac de données, puis dans plusieurs éléments du sac de données pour écrire des messages dans le journal Chef avec les noms abrégés et la source URLs des applications :

```
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
déploiement	attributs	domains
enable_ssl	environment	name
shortname	ssl_configuration	type

app_id

ID d'application (chaîne). GUID qui identifie l'application.

app_source

Ensemble de contenu qui spécifie les informations que AWS OpsWorks Stacks utilise pour déployer l'application à partir de son référentiel de contrôle de source. Le contenu varie en fonction du type de référentiel.

mot de passe

Mot de passe pour les référentiels privés et `"null"` pour les référentiels publics (chaîne). Pour les compartiments S3 privés, le contenu est défini sur la clé secrète.

révision

Si le référentiel comporte plusieurs branches, le contenu spécifie la branche ou la version de l'application, telle que `"version1"` (chaîne). Sinon, elle est définie sur `"null"`.

ssh_key

Clé SSH de déploiement pour accéder aux référentiels Git privés, et `"null"` pour les référentiels publics (chaîne).

type

Emplacement source de l'application (chaîne). Les valeurs valides sont les suivantes :

- `"archive"`
- `"git"`
- `"other"`
- `"s3"`

url

Où se trouve la source de l'application (chaîne).

utilisateur

Nom d'utilisateur pour les référentiels privés et `"null"` pour les référentiels publics (chaîne). Pour les compartiments S3 privés, le contenu est défini sur la clé d'accès.

attributs

Ensemble de contenus qui décrit la structure du répertoire et le contenu de l'application.

document_root

Répertoire racine de l'arborescence de document. Définit le chemin d'accès à la racine du document (ou l'emplacement de la page d'accueil de l'application), tel que `home_html`, qui est relatif au répertoire de votre déploiement. Sauf si cet attribut est spécifié, la valeur par défaut de document_root est `public`. La valeur de `document_root` peut commencer uniquement par les caractères `a-z`, `A-Z`, `0-9`, `_` (trait de soulignement) ou `-` (tiret).

data_sources

Informations requises pour se connecter à la base de données de l'application. Si une couche de base de données est attachée à l'application, AWS OpsWorks Stacks attribue automatiquement les valeurs appropriées à ce contenu.

La valeur de data_sources est un tableau, et les tableaux sont accessibles par un décalage de type par entier, pas par clé. Par exemple, pour accéder à la première source des données de l'application, utilisez `app[:data_sources][0][:type]`.

database_name

Nom de la base de données, qui est généralement le nom court de l'application (chaîne).

type

Type de l'instance de base de données, généralement `"RdsDbInstance"` (chaîne).

arn

ARN (Amazon Resource Name) de l'instance de base de données (chaîne).

déploiement

Si l'application doit être déployée (booléen). `true` pour les applications qui doivent être déployées dans un événement de cycle de vie Deploy. Dans un événement de cycle de vie Deploy, le contenu sera `true` pour toutes les applications. Pour déterminer quelles applications doivent être déployées sur une instance, vérifiez les couches auxquelles l'instance appartient.

domains

Liste des domaines de l'application (liste de chaînes).

enable_ssl

Indique si la prise en charge SSL est activée (valeur booléenne).

environment

Ensemble de variables d'environnement spécifiées par l'utilisateur qui ont été définies pour l'application. Pour plus d'informations sur la définition des variables d'environnement d'une application, consultez Ajout d'applications. Chaque nom de contenu est défini sur un nom de variable environnement et la valeur correspondante est définie sur la valeur de la variable.

name

Nom de l'application, utilisé à des fins d'affichage (chaîne).

shortname

Le nom abrégé de l'application, généré par AWS OpsWorks Stacks à partir du nom (chaîne). Le nom court est utilisé en interne et par les recettes ; il sert de nom pour le répertoire où les fichiers de votre application sont installés.

ssl_configuration

certificate

Si vous avez activé la prise en charge SSL, le certificat SSL de l'application ; sinon, `"null"` (chaîne).

chain

Si SSL est activé, le contenu pour spécifier une clé d'autorité de certification intermédiaire ou une authentification client (chaîne).

private_key

Si vous avez activé la prise en charge SSL, la clé privée SSL de l'application ; sinon, `"null"` (chaîne).

type

Type de l'application, qui est toujours défini comme `"other"` pour les piles Linux Chef 12 et les piles Windows 12.2 Chef (chaîne).

Conteneurs de données de commandes (aws_opsworks_command)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Représente les paramètres d'une commande exécutée par AWS OpsWorks Stacks sur une ou plusieurs instances.

L'exemple suivant montre comment utiliser la recherche Chef pour effectuer une recherche via un seul élément de conteneur de données, puis via plusieurs conteneurs de données pour écrire les messages dans le journal Chef avec les types de commandes et l'emplacement où ils ont été envoyés :

```
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	type

args

Arguments de la commande (chaîne).

command_id

Identifiant unique aléatoire de la commande, attribué par AWS OpsWorks Stacks (chaîne).

iam_user_arn

Si la commande est créée par le client, l'ARN de l'utilisateur qui a créé la commande (chaîne).

instance_id

Identifiant de l'instance sur laquelle la commande a été exécutée (chaîne).

sent_at

Horodatage du moment où AWS OpsWorks Stacks a exécuté la commande (chaîne).

type

Type de la commande (chaîne). Les valeurs valides sont les suivantes :

- `"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"

Sac de données du cluster Amazon ECS (aws_opsworks_ecs_cluster)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Représente les paramètres d'un cluster Amazon ECS.

L'exemple suivant montre comment utiliser la recherche Chef pour rechercher dans un seul élément de sac de données, puis dans plusieurs éléments de sac de données pour écrire des messages dans le journal Chef avec les noms des clusters Amazon ECS et les noms des ressources Amazon () ARNs :

```
ecs_cluster = search("aws_opsworks_ecs_cluster").first
Chef::Log.info("********** The ECS cluster's name is
 '#{ecs_cluster['ecs_cluster_name']}' **********")
Chef::Log.info("********** The ECS cluster's ARN is '#{ecs_cluster['ecs_cluster_arn']}'
 **********")

search("aws_opsworks_ecs_cluster").each do |ecs_cluster|
  Chef::Log.info("********** The ECS cluster's name is
 '#{ecs_cluster['ecs_cluster_name']}' **********")
```

```
  Chef::Log.info("********** The ECS cluster's ARN is
 '#{ecs_cluster['ecs_cluster_arn']}' **********")
end
```

ecs_cluster_arn	ecs_cluster_name

ecs_cluster_arn

ARN du cluster (chaîne).

ecs_cluster_name

Nom du cluster (chaîne).

Sac de données Elastic Load Balancing (aws_opsworks_elastic_load_balancer)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Représente les paramètres d'un équilibreur de charge Elastic Load Balancing.

L'exemple suivant montre comment utiliser la recherche Chef pour rechercher dans un seul élément de sac de données, puis dans plusieurs éléments de sac de données pour écrire des messages dans le journal Chef avec les noms des équilibreurs de charge et les noms DNS des équilibreurs de charge Elastic Load Balancing :

```
elastic_load_balancer = search("aws_opsworks_elastic_load_balancer").first
Chef::Log.info("********** The ELB's name is
 '#{elastic_load_balancer['elastic_load_balancer_name']}' **********")
Chef::Log.info("********** The ELB's DNS name is '#{elastic_load_balancer['dns_name']}'
 **********")
```

```
search("aws_opsworks_elastic_load_balancer").each do |elastic_load_balancer|
  Chef::Log.info("********** The ELB's name is
 '#{elastic_load_balancer['elastic_load_balancer_name']}' **********")
  Chef::Log.info("********** The ELB's DNS name is
 '#{elastic_load_balancer['dns_name']}' **********")
end
```

elastic_load_balancer_name	dns_name	layer_id

elastic_load_balancer_name

Nom de l'équilibreur de charge (chaîne).

dns_name

Nom DNS de l'équilibreur de charge (chaîne).

layer_id

ID de AWS OpsWorks pile de la couche à laquelle l'équilibreur de charge est attribué (chaîne).

Conteneur de données de l'instance (aws_opsworks_instance)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Représente les paramètres d'une instance.

L'exemple suivant montre comment utiliser la recherche Chef pour rechercher dans un seul élément du sac de données, puis dans plusieurs éléments du sac de données pour écrire des messages dans le journal Chef avec les noms d'hôte des instances et : IDs

```
instance = search("aws_opsworks_instance").first
```

```
Chef::Log.info("********** The instance's hostname is '#{instance['hostname']}'
 **********")
Chef::Log.info("********** The instance's ID is '#{instance['instance_id']}'
 **********")

search("aws_opsworks_instance").each do |instance|
  Chef::Log.info("********** The instance's hostname is '#{instance['hostname']}'
 **********")
  Chef::Log.info("********** The instance's ID is '#{instance['instance_id']}'
 **********")
end
```

L'exemple suivant montre différentes manières d'utiliser la recherche Chef pour rechercher dans plusieurs éléments du sac de données afin de trouver l'élément du sac de données contenant l'ID d' EC2instance Amazon spécifié. L'exemple utilise ensuite le contenu de l'élément de conteneur de données sac pour écrire un message dans le journal Chef avec l'adresse IP publique de l'instance correspondante :

```
instance = search("aws_opsworks_instance", "ec2_instance_id:i-12345678").first
Chef::Log.info("********** For instance '#{instance['ec2_instance_id']}', the
 instance's public IP address is '#{instance['public_ip']}' **********")

search("aws_opsworks_instance").each do |instance|
  if instance['ec2_instance_id'] == 'i-12345678'
    Chef::Log.info("********** For instance '#{instance['ec2_instance_id']}', the
 instance's public IP address is '#{instance['public_ip']}' **********")
  end
end
```

L'exemple suivant montre comment utiliser la recherche de Chef avec `self:true` pour trouver l'élément de conteneur de données qui contient les informations associées à l'instance sur laquelle la recette est en cours d'exécution. L'exemple utilise ensuite le contenu de l'élément du sac de données pour écrire un message dans le journal Chef avec l'ID AWS OpsWorks généré par Stacks de l'instance correspondante et l'adresse IP publique de l'instance :

```
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	application	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

ID d'AMI (Amazon Machine Image) de l'instance (chaîne).

application

Architecture de l'instance, qui est toujours définie sur `"x86_64"` (chaîne).

auto_scaling_type

Type de dimensionnement de l'instance : `null`, `timer` ou `load` (chaîne).

availability_zone

Zone de disponibilité de l'instance, telle que `"us-west-2a"` (chaîne).

created_at

Date à laquelle l'instance a été créée, au format UTC "*yyyy*-*mm*-*ddd*T*hh*:*mm*:*ss*+*hh*:*mm*" (chaîne). Par exemple, `"2013-10-01T08:35:22+00:00"` correspond à 08:35:22 le 10 octobre 2013, sans décalage horaire. Pour plus d'informations, consultez ISO 8601.

ebs_optimized

Indique si l'instance est optimisée pour les volumes EBS (valeur booléenne).

ec2_instance_id

ID de l' EC2 instance (chaîne).

elastic_ip

Adresse IP Elastic ; définie sur `"null"` si l'instance n'a pas d'adresse IP Elastic (chaîne).

hostname

Nom d'hôte, tel que `"demo1"` (chaîne).

instance_id

L'ID d'instance, qui est un GUID AWS OpsWorks généré par Stacks qui identifie l'instance de manière unique (chaîne).

instance_type

Type de l'instance, tel que `"c1.medium"` (chaîne).

layer_ids

Liste des couches de l'instance, identifiées par leur caractère unique IDs ; par exemple,`307ut64c-c7e4-40cc-52f0-67d5k1f9992c`.

os

Système d'exploitation de l'instance (chaîne). Les valeurs valides sont les suivantes :

- `"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

Nom DNS privé (chaîne).

private_ip

Adresse IP privée (chaîne).

public_dns

Nom DNS public (chaîne).

public_ip

Adresse IP publique (chaîne).

root_device_type

Type d'appareil racine (chaîne). Les valeurs valides sont les suivantes :

- "ebs
- "instance-store"

root_device_volume_id

ID de volume de l'appareil racine (chaîne).

self

`true`Si cet élément de conteneur de données contient des informations sur l'instance sur laquelle la recette est en cours d'exécution ; sinon, `false` (valeur booléenne). Cette valeur n'est disponible que pour les recettes, et non via l'API AWS OpsWorks Stacks.

ssh_host_dsa_key_fingerprint

Séquence plus courte d'octets qui identifie la clé publique DSA plus longue (chaîne).

ssh_host_dsa_key_private

Clé privée générée par DSA pour l'authentification SSH auprès de l'instance (chaîne).

ssh_host_dsa_key_public

Clé publique générée par DSA pour l'authentification SSH auprès de l'instance (chaîne).

ssh_host_rsa_key_fingerprint

Séquence plus courte d'octets qui identifie la clé publique RSA plus longue (chaîne).

ssh_host_rsa_key_private

Clé privée générée par RSA pour l'authentification SSH auprès de l'instance (chaîne).

ssh_host_rsa_key_public

Clé publique générée par RSA pour l'authentification SSH auprès de l'instance (chaîne).

status

État de l'instance (chaîne). Les valeurs valides sont les suivantes :

- `"requested"`
- `"booting"`
- `"running_setup"`
- `"online"`
- `"setup_failed"`
- `"start_failed"`
- `"terminating"`
- `"terminated"`
- `"stopped"`
- `"connection_lost"`

subnet_id

ID de sous-réseau de l'instance (chaîne).

virtualization_type

Type de virtualisation de l'instance (chaîne).

Conteneur de données de la couche (aws_opsworks_layer)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Représente les paramètres d'une couche.

L'exemple suivant montre comment utiliser la recherche Chef pour effectuer une recherche via un seul élément de conteneur de données, puis via plusieurs éléments pour écrire les messages dans le journal Chef avec les noms des couches et les noms courts :

```
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
packages	shortname	type
volume_configurations		

ecs_cluster_arn

Si un cluster Amazon ECS est attribué à la couche, le nom de ressource Amazon (ARN) (chaîne) du cluster Amazon ECS.

chiffrées

`true` si le volume EBS est chiffré ; sinon, `false` (valeur booléenne).

layer_id

L'ID de couche, qui est un GUID généré par AWS OpsWorks Stacks et qui identifie de manière unique la couche (chaîne).

name

Nom de la couche, utilisé pour représenter la couche dans la console (chaîne). Il peut être défini par l'utilisateur et ne doit pas être unique.

packages

Liste de packages à installer (liste de chaînes).

shortname

Nom court de la couche, défini par l'utilisateur (chaîne).

type

Type de la couche, qui est toujours défini comme `"custom"` pour Linux Chef 12 et Windows 12.2 Chef (chaîne).

volume_configurations

Liste des configurations de volumes Amazon EBS.

iops

Nombre d'opérations d'E/S par seconde que le volume peut prendre en charge.

mount_point

Répertoire du point de montage du volume.

number_of_disks

Nombre de disques du volume.

raid_level

Niveau de configuration RAID du volume.

size

Taille du volume en gibioctets (Gio).

volume_type

Type de volume : à usage général, magnétique, à IOPS provisionnées, avec HDD à débit optimisé ou HDD à froid.

Sac de données Amazon RDS (aws_opsworks_rds_db_instance)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Ensemble de contenu de sac de données qui spécifie la configuration d'une instance Amazon Relational Database Service (Amazon RDS) comme suit :

adresse	db_instance_identifier	db_password
db_user	engine	rds_db_instance_arn
region		

L'exemple suivant montre comment utiliser la recherche Chef pour rechercher dans un seul élément de sac de données, puis dans plusieurs éléments de sac de données pour écrire des messages dans le journal Chef avec les adresses et les types de moteurs de base de données des instances Amazon RDS :

```
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
```

adresse

Nom DNS de l'instance.

port

Port de l'instance.

db_instance_identifier

ID d'instance.

db_password

Mot de passe principal de l'instance.

db_user

Nom d'utilisateur principal de l'instance.

engine

Moteur de base de données de l'instance, tel que `mysql`.

rds_db_instance_arn

ARN de l'instance.

region

Région AWS de l'instance, telle que `us-west-2`.

Conteneur de données de la pile (aws_opsworks_stack)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement

aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Représente les paramètres d'une pile.

L'exemple suivant montre comment utiliser la recherche Chef pour écrire des messages dans le journal Chef avec le nom de la pile et l'url source du livre de recettes :

```
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

ARN de la pile (chaîne).

custom_cookbooks_source

Ensemble de contenus qui spécifient le référentiel source du livre de recettes personnalisé.

type

Type de référentiel (chaîne). Les valeurs valides sont les suivantes :

- `"archive"`
- `"git"`
- `"s3"`

url

URL de référentiel, telle que `"git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git"` (chaîne).

nom d'utilisateur

Nom d'utilisateur pour les référentiels privés et `null` pour les référentiels publics (chaîne). Pour les compartiments privés Amazon Simple Storage Service (Amazon S3), le contenu est défini sur la clé d'accès.

mot de passe

Mot de passe pour les référentiels privés et `null` pour les référentiels publics (chaîne). Pour les compartiments S3 privés, le contenu est défini sur la clé secrète.

ssh_key

Clé SSH de déploiement pour accéder aux référentiels Git privés, et `null` pour les référentiels publics (chaîne).

révision

Si le référentiel comporte plusieurs branches, le contenu spécifie la branche ou la version de l'application, telle que `"version1"` (chaîne). Sinon, il est défini comme `null`.

name

Nom de la pile (chaîne).

region

Région AWS de la pile (chaîne).

stack_id

GUID qui identifie la pile (chaîne).

use_custom_cookbooks

Indique si les livres personnalisés sont activés (valeur booléenne).

vpc_id

Si la pile est en cours d'exécution dans un VPC, l'ID de VPC (chaîne).

Conteneur de données utilisateur (aws_opsworks_user)

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement

aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Représente les paramètres d'un utilisateur.

L'exemple suivant montre comment utiliser la recherche Chef pour rechercher dans un seul élément du sac de données, puis dans plusieurs éléments du sac de données pour écrire des messages dans le journal Chef avec les noms d'utilisateur des utilisateurs et les noms des ressources Amazon (ARNs) :

```
user = search("aws_opsworks_user").first
Chef::Log.info("********** The user's user name is '#{user['username']}' **********")
Chef::Log.info("********** The user's user ARN is '#{user['iam_user_arn']}'
 **********")

# Or...

search("aws_opsworks_user").each do |user|
  Chef::Log.info("********** The user's user name is '#{user['username']}' **********")
  Chef::Log.info("********** The user's user ARN is '#{user['iam_user_arn']}'
 **********")
end
```

administrator_privileges	iam_user_arn	remote_access
ssh_public_key	unix_user_id	nom d'utilisateur

administrator_privileges

Indique si l'utilisateur a des privilèges d'administrateur (valeur booléenne).

iam_user_arn

ARN de l'utilisateur (chaîne).

remote_access

Indique si l'utilisateur peut utiliser RDP pour se connecter à l'instance (valeur booléenne).

ssh_public_key

La clé publique de l'utilisateur, telle que fournie via la console ou l'API AWS OpsWorks Stacks (chaîne).

unix_user_id

ID Unix de l'utilisateur (nombre).

nom d'utilisateur

Nom d'utilisateur (chaîne).

OpsWorks Changements d'agent

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Versions de l'agent Chef 12

Le tableau suivant décrit les modifications importantes apportées à l'agent Chef 12 qu'AWS OpsWorks Stacks installe sur les instances qu'il gère.

Version d'agent	Description	Date de parution
4042	• Cette version de l'agent ne contient que des modifications mineures, sans nouvelles fonctionn alités.	7 février 2023
4041	• Cette version de l'agent ne contient que des modifications mineures, sans nouvelles fonctionn alités. • Mettre à jour les certificats Amazon CA	27 janvier 2023

Version d'agent	Description	Date de parution
4040	• Cette version de l'agent ne contient que des modifications mineures, sans nouvelles fonctionn alités.	22 juillet 2022
4039	• Corriger l'intégration ECS pour Ubuntu AMIs	30 avril 2020
4038	• Correction d'un bogue lors de l'envoi de statistiques d'instance pendant le passage à l'heure d'été • Respecter la variable d'environnement `no_proxy` pendant le téléchargement et l'installation de l'agent	5 mars 2020
4037	• Ajout de la prise en charge de la signature de demandes vers S3 URLs sans région à l'aide de SigV4 • Suppression de la prise en charge de la signature des requêtes S3 à l'aide de SigV2	4 juin 2019
4035	• Correction d'un bogue lors de l'installation d'ECS • Correction des entrées `fstab` dupliquées après la modification d'un type d'instance	8 mai 2019
4033	• Ajout de la prise en charge d'Ubuntu 18.04 • Correction d'un bogue d'installation de l'agent dans Amazon Linux 2	26 novembre 2018
4032	• Ajout de la prise en charge d'Amazon Linux 2	24 octobre 2018
4031	• Ajout de la prise en charge d'Amazon Linux 2018.03 • Prise en charge des archives S3 publiques hébergées sur un autre compte	15 août 2018
4030	• Correction de la gestion des volumes pour les instances c5d	31 mai 2018

Version d'agent	Description	Date de parution
4029	• Installation de `nvme-cli` sur Ubuntu 14.04 • Correction du montage de volume sur les instances c5, m5 • Conservation systématique du nom d'hôte au redémarrage	2 mai 2018
4028	• Correction de la configuration de `monit` pour CentOS	le 20 mars 2018
4027	• Support pour le montage de NVMe volumes sur Ubuntu 14.04 (`nvme-cli`doit être installé manuellement) • La propriété `name` n'est pas exigée pour les volumes	17 février 2018
4026	• Montage NVMe de volumes EBS basés sur l'ID de volume EBS • Correction du montage de volume EBS sur les instances i3 • Correction de l'ordre des volumes EBS montés sur les instances c5, m5	31 janvier 2018
4025	• Corriger la manipulation des NVMe appareils	le 13 décembre 2017
4024	• Ajout de la prise en charge d'Amazon Linux 2017.09	5 décembre 2017
4023	• Ajout de la prise en charge de l'intégration CloudWatch des journaux	2 avril 2017
4022	• Mise à jour du client Chef vers la version 12.18.31	1 février 2017
4021	• Amélioration de la gestion des proxys	le 16 décembre 2016
4020	• Mise à jour du client Chef vers la version 12.16.42	8 décembre 2016

Version d'agent	Description	Date de parution
4019	• Variables du proxy source lors de l'installation de l'agent • Red Hat Enterprise Linux 7 utilise désormais `systemd` au lieu de `monit` • Ne configurez pas EPEL sur Red Hat Enterprise Linux 7 • Utilisez `flock` au lieu de `lockrun.c` pour le verrouillage de processus. • Evitez les sorties bizarres de `ps -p1` lors du contrôle de `systemd`	19 octobre 2016
4018	• Mise à jour du client Chef vers la version 12.13.37 • Ajout de la prise en charge d'Amazon Linux 2016.09	25 août 2016
4017	• Mise à jour du client Chef vers la version 12.12.15	10 août 2016
4016	• Correction de la désinstallation de l'agent sur les systèmes où `monit` n'est pas utilisé	23 juin 2016
4015	• Correction de l'installation d'ECS pour Amazon Linux 2016.03	17 juin 2016
4011	• Mise à jour du client Chef vers la version 12.10.24 • Amélioration de la gestion du chargement des journaux	19 mai 2016
4008	• Ajout de la prise en charge d'Amazon Linux 2016.03 • Ajout du délai d'installation groupée • Ajoutez xfs to /etc/filesystems s'il existe	16 mars 2016

Version d'agent	Description	Date de parution
4007	• Mise à jour du client Chef vers la version 12.7.2 • Améliorations de la gestion des erreurs pour les instances locales (serveurs hébergés à l'extérieur d'AWS) • Amélioration de la compatibilité avec la version chef-sugar la plus récente • Nouvelle tentative de téléchargement d'archive pour le déploiement	4 mars 2016
4006	• Mise à jour du client Chef vers la version 12.6.0 • N'installez pas les paquets libxml2- devel/libxml2-dev and libxslt-devel/libxslt -dev lors de l'installation de l'agent	21 janvier 2016
4005	• Correction de l'importation ec2 en autorisant toujours les données ec2 dans ohai pour l'infrast ructure ec2	17 décembre 2015
4004	• AWS OpsWorks Support de Stacks pour Chef 12 Linux- Chef Client 12.5.1	3 décembre 2015

Versions de l'agent Chef 11.10

Le tableau suivant décrit les modifications importantes apportées à l'agent Chef 11.10 qu'AWS OpsWorks Stacks installe sur les instances qu'il gère.

Version d'agent	Description	Date de parution
3456	• Cette version de l'agent ne contient que des modifications mineures, sans nouvelles fonctionn alités. • Mettre à jour les certificats Amazon CA	27 janvier 2023

Version d'agent	Description	Date de parution
3455	• Cette version de l'agent ne contient que des modifications mineures, sans nouvelles fonctionn alités.	1er novembre 2022
3454	• Corriger l'intégration ECS pour Ubuntu AMIs	28 avril 2020
3453	• Correction d'un bogue lors de l'envoi de statistiques d'instance pendant le passage à l'heure d'été • Correction d'un bogue de paquets manquants dans l' RHEL7 installation • Respecter la variable d'environnement `no_proxy` pendant le téléchargement et l'installation de l'agent	5 mars 2020
3452	• N'incluez pas la région dans l'URL du chemin virtuel Amazon S3 si elle est `us-east-1` • Extraire et télécharger des livres de cuisine internes dans des compartiments spécifiques à la région d'étape • Corriger les entrées `fstab` pour Chef 11.10 • Supprimer l'utilisation de SigV2 pour S3 et obtenir la région pour le compartiment dans la demande	13 août 2019
3451	• Ajout de la prise en charge de Ruby 2.6.1	20 mars 2019
3450	• Correction des attributs EBS par défaut • Installation de CloudWatchLogs l'agent Corriger pour Amazon Linux 2 • Correction de l'installation de bundler pour les versions de rubygem plus récentes que la version 2.6.14 • Correction de la prise en charge des archives S3 publiques	3 décembre 2018

Version d'agent	Description	Date de parution
3449	• Correction de la gestion des volumes pour les instances c5d • Corriger la prise en charge des matrices RAID sur les NVMe instances -device	5 juin 2018
3448	• Mise à niveau de la version 2.3 par défaut de Ruby vers la version 2.3.7 • Corriger le montage de volumes EBS sur des instances NVMe basées sur des instances Ubuntu 14.04 • Support des archives publiques Amazon S3 hébergées sur un autre compte • Résolution des problèmes de démarrage d'`opsworks-agent` sur les instances Red Hat Enterprise Linux	8 mai 2018
3447	• Montage NVMe de volumes EBS basés sur l'ID de volume EBS • Correction du montage de volume EBS sur les instances i3 • Correction de l'ordre des volumes EBS montés sur c5, m5 • Mise à jour de la version 2.3 par défaut de Ruby vers la version 2.3.6	31 janvier 2018
3446	• Corriger la manipulation des NVMe appareils • Mise à jour de la version 2.3 par défaut de Ruby vers la version 2.3.5	14 décembre 2017
3445	• Ajout de la prise en charge d'Amazon Linux 2017.09 • Mise à jour de la version 2.2 par défaut de Ruby vers la version 2.2.8	31 octobre 2017
3444	• Ajouter le support pour les CloudWatch journaux	1 avril 2017

Version d'agent	Description	Date de parution
3443	• Amélioration de la gestion des proxys	le 15 décembre 2016
3442	• Mise à jour de la version 2.3 par défaut de Ruby vers la version 2.3.3 • Mise à jour de la version 2.2 par défaut de Ruby vers la version 2.2.6	6 décembre 2016
3441	• Variables du proxy source lors de l'installation de l'agent	21 octobre 2016
3440	• Ajout de la prise en charge d'Amazon Linux 2016.09	13 septembre 2016
3439	• Modifications mineures ; aucune nouvelle fonction	29 juillet 2016
3438	• Ajout de la prise en charge de Ruby 2.3.1 • Amélioration de l'enregistrement des instances avec les informations d'identification du profil d'instance IAM • Suppression des reliquats `s3curl.pl` • Correction de l'installation d'ECS pour Amazon Linux 2016.03	17 juin 2016
3437	• Mise à jour de la version 2.2 par défaut de Ruby vers la version 2.2.5	4 mai 2016
3436	• Mise à jour de l'URL EPEL pour Red Hat Enterpris e Linux. IMPORTANT : sans cette modification, le démarrage des instances Red Hat Enterprise Linux échoue.	18 avril 2016
3435	• Mise à jour de la version 2.1 par défaut de Ruby vers la version 2.1.9 • Améliorez la gestion d'Amazon S3 et des déploieme nts d'archives	6 avril 2016

Version d'agent	Description	Date de parution
3434	• Ajout de la prise en charge d'Amazon Linux 2016.03 • Nouvelle tentative d'installation des packages	16 mars 2016
3433	• Quelques améliorations pour les instances sur site (serveurs hébergés en dehors de AWS) • Amélioration de la compatibilité avec la version la plus récente de `chef-sugar` • Nouvelle tentative de téléchargement d'archive pour le déploiement • Correction de l'URL d'installation de gems Ruby	27 février 2016
3432	• Amélioration de la gestion des caractères spéciaux dans les noms de compartiment • Mise à jour de `s3_file` vers la version 2.6.6 • Ignorer le montage des volumes sans point de montage spécifié • Redémarrage systématique d'`unicorn` au lieu d'arrêter et de redémarrer afin d'éviter les temps d'arrêt pendant le déploiement • Mise à jour systématique du livre de recettes personnalisé pour la commande `setup` • Après la création de matrices RAID, mise à jour d'`initramfs` pour éviter les problèmes de mappage de périphérique lors du redémarrage	20 janvier 2016
3431	• Résolution du problème d'installation des gems `passenger` et `unicorn` dans la couche Rails • Mise à jour des versions 2.0, 2.1 et 2.2 par défaut de Ruby vers les versions 2.0.0p648, 2.1.8 et 2.2.4 • Autorisation de la définition de noms de package `postgres` dans le JSON personnalisé • Mise à jour de la version par défaut de Node.js vers la version 0.12.9	22 décembre 2015

Version d'agent	Description	Date de parution
3430	• Modifications mineures ; aucune nouvelle fonction	25 novembre 2015
3429	• Améliorer le daemonize des OpsWorks agents (fermer stdout/stderr) • Amélioration de la robustesse de la ressource `s3_file` (nouvelles tentatives, exceptions intercept ées)	18 novembre 2015
3428	• L'ajout d'une détection d'`postgres`adaptateur basée sur le Gemfile corrige https://github.com/aws/opsworks-cookbooks/issues/136	17 juin 2016
3427	• Résolution d'un problème lié à la récupération des informations d'identification de l'agent • Mise à jour des versions 2.0, 2.1 et 2.2 par défaut de Ruby vers les versions 2.0.0p647, 2.1.7 et 2.2.3	11 septembre 2015
3426	• Mise à jour d'`aws-sdk` vers la version 1.65.0 • Améliorer le téléchargement depuis Amazon S3 en remplaçant `s3curl` par `s3_file cookbook` • Remplacement de la version par défaut de Node.js par la version 0.12.7 • Ajout de la journalisation pour les applications Node.js. STDOUT et STDERR sont consignés et font l'objet d'une rotation dans le répertoire `shared/log` • La mise à jour de l'extraction de sous-module de livre de recettes personnalisé a été rendue explicite • Ajout d'une solution de contournement pour https://github.com/aws/ opsworks-cookbooks/issues/213 qui vérifiera que les montages de liaison ont été effectués avant la création du répertoire `deploy`	27 août 2015
3425	• Prise en charge par ECS d'Amazon Linux et Ubuntu	27 juillet 2015

Version d'agent	Description	Date de parution
3424	• Modifications mineures ; aucune nouvelle fonction	9 juillet 2015
3422	• Prise en charge complète de Red Hat Enterprise Linux 7 • Renforcement de la résistance aux erreurs de la génération de `/etc/hosts`	29 juin 2015
3421	• Option de remplacement du nom de package de base de données pour Red Hat Enterprise Linux 7 • Mise à jour de la configuration de `monit systemd` pour éviter que `systemd` envoie le signal `kill` aux processus surveillés par `monit`.	11 juin 2015

AWS OpsWorks Ressources Stacks

> **Important**
>
> Le AWS OpsWorks Stacks service a atteint sa fin de vie le 26 mai 2024 et a été désactivé tant pour les nouveaux clients que pour les clients existants. Nous recommandons vivement aux clients de migrer leurs charges de travail vers d'autres solutions dès que possible. Si vous avez des questions sur la migration, contactez l' AWS Support équipe sur AWS Re:Post ou via le AWS Support Premium.

Les ressources connexes suivantes peuvent s'avérer utiles lors de l'utilisation de ce service.

Guides de référence, outils et ressources de prise en charge

Plusieurs guides, forums, coordonnées et autres ressources utiles sont disponibles à partir d' AWS OpsWorks Stacks et d'Amazon Web Services.

- AWS OpsWorks Référence de l'API Stacks — Descriptions, syntaxe et exemples d'utilisation des actions et des types de données de AWS OpsWorks Stacks, y compris les paramètres courants et les codes d'erreur.
- AWS OpsWorks FAQ technique Stacks — Questions les plus fréquemment posées par les développeurs à propos de ce produit.
- AWS OpsWorks Notes de mise à jour de Stacks — Présentation générale de la version actuelle. Ce document indique, en particulier, les nouvelles fonctions et dernières corrections apportées, ainsi que les problèmes connus.
- Outils AWS pour PowerShell : ensemble d' PowerShell applets de commande Windows qui présentent les fonctionnalités de AWS SDK pour .NET l' PowerShell environnement.
- Interface de ligne de commande AWS : syntaxe de ligne de commande uniforme pour accéder aux services AWS. L'interface de ligne de commande AWS utilise un processus d'installation unique pour autoriser l'accès à tous les services pris en charge.
- AWS OpsWorks Référence de ligne de commande Stacks : commandes AWS OpsWorks spécifiques à Stacks à utiliser à l'invite de ligne de commande.

- Cours et ateliers — Liens vers des cours spécialisés et basés sur des rôles, ainsi que des ateliers à votre rythme pour vous aider à perfectionner vos AWS compétences et à acquérir une expérience pratique.
- AWS Centre pour développeurs : découvrez les didacticiels, téléchargez des outils et découvrez les événements AWS destinés aux développeurs.
- AWS Outils de développement : liens vers des outils de développement SDKs, des boîtes à outils IDE et des outils de ligne de commande pour le développement et la gestion d' AWS applications.
- Centre de ressources pour la mise en route : découvrez comment configurer votre application Compte AWS, rejoindre la AWS communauté et lancer votre première application.
- Tutoriels pratiques — Suivez les step-by-step didacticiels pour lancer votre première application sur AWS.
- AWS Livres blancs : liens vers une liste complète de livres AWS blancs techniques, traitant de sujets tels que l'architecture, la sécurité et l'économie, rédigés par des architectes de AWS solutions ou d'autres experts techniques.
- AWS Support Centre — Le centre de création et de gestion de vos AWS Support dossiers. Comprend également des liens vers d'autres ressources utiles, telles que des forums, des informations techniques FAQs, l'état de santé des services et AWS Trusted Advisor.
- Support— La principale page Web contenant des informations sur Support un one-on-one canal d'assistance à réponse rapide pour vous aider à créer et à exécuter des applications dans le cloud.
- Contactez-nous : point de contact central pour toute question relative à la facturation AWS , à votre compte, aux événements, à des abus ou à d'autres problèmes.
- AWS Conditions du site — Informations détaillées sur nos droits d'auteur et notre marque commerciale ; votre compte, votre licence et l'accès au site ; et d'autres sujets.

AWS Kits de développement logiciel

Amazon Web Services fournit des kits de développement logiciel permettant d'accéder à AWS OpsWorks Stacks à partir de différents langages de programmation. Les bibliothèques SDK automatisent un certain nombre de tâches courantes, telles que la signature chiffrée des demandes de service, les nouvelles tentatives et la gestion des réponses erronées.

- AWS SDK pour Java— Configuration et autres documentations
- AWS SDK pour .NET— Configuration et autres documentations.
- Kit de développement logiciel AWS pour PHP — Documentation

- AWS SDK pour Ruby— Documentations
- autre documentation
- AWS SDK for Python (Boto)— Configuration et autres documentations

Logiciels Open Source

AWS OpsWorks Stacks inclut une variété de progiciels open source, régis par leurs licences respectives. Pour plus d'informations, consultez les ressources suivantes :

- Pour les instances Linux Chef 12, ouvrez le fichier `THIRD_PARTY_LICENSES` du répertoire `/opt/aws/opsworks/current` de l'instance.
- Pour Chef 11.10 et versions antérieures pour Linux, téléchargez le document PDF sur les attributions des agents OpsWorks Linux.

AWS OpsWorks historique du document

Modification	Description	Date
Mises à jour de AWS OpsWorks Stacks	Vous pouvez désormais utiliser l'outil Detach in Place pour détacher vos OpsWorks instances du service OpsWorks Stacks, voir Utilisation de l'outil AWS OpsWorks Stacks Detach in Place dans ce guide.	11 avril 2024
Mises à jour de AWS OpsWorks Stacks	Vous pouvez désormais migrer vos AWS OpsWorks Stacks vers AWS Systems Manager Application Manager à l'aide d'un script de migration . Pour plus d'informations, consultez la section Migration de vos AWS OpsWorks Stacks applications vers AWS Systems Manager Application Manager dans ce guide.	22 décembre 2022
Mises à jour de AWS OpsWorks for Chef Automate et AWS OpsWorks for Puppet Enterprise	Une procédure de dépannage est désormais disponible qui décrit ce que vous pouvez faire en cas d'échec de la maintenance du système pour votre serveur AWS OpsWorks for Chef Automate ou OpsWorks pour Puppet Enterprise. Pour plus d'informations, voir Échec de la maintenance du système pour	29 septembre 2022

	le serveur Chef Automate ou Échec de la maintenance du système pour le serveur Puppet Enterprise. dans ce guide.	
Mises à jour de AWS OpsWorks for Chef Automate et AWS OpsWorks for Puppet Enterprise	Une procédure de dépannage est désormais disponibl e si votre serveur AWS OpsWorks for Chef Automate ou celui OpsWorks de Puppet Enterprise entre dans un `Connection lost` état. Pour plus d'informations, voir le serveur Chef Automate est dans un `Connectio n lost` état ou le serveur Puppet Enterprise est dans un `Connection lost` état dans ce guide.	23 mars 2022
Mises à jour de AWS OpsWorks Stacks	Pour des raisons de sécurité, vous pouvez désormais ajouter une clé de `aws:Sourc eAccount` condition `aws:SourceArn` ou une clé de condition (ou les deux) aux politiques de relation de confiance qui permetten t à AWS OpsWorks Stacks d'accéder à des tâches dans d'autres AWS services. Pour plus d'informations, consultez la section Prévention de la confusion entre les services adjoints dans AWS OpsWorks Stacks dans ce guide.	4 mars 2022

Mises à jour de AWS OpsWorks for Chef Automate et AWS OpsWorks for Puppet Enterprise	Pour des raisons de sécurité, vous pouvez désormais ajouter une clé `aws:SourceArn` ou une clé de `aws:SourceAccount` condition (ou les deux) aux politiques de relation de confiance qui autorisent AWS OpsWorks for Chef Automate et permettent OpsWorks à Puppet Enterprise d'accéder à des tâches dans d'autres AWS services. Pour plus d'informations, consultez la section Prévention interservices de la confusion des adjoints dans ce guide.	10 janvier 2022
Mises à jour de AWS OpsWorks for Chef Automate et AWS OpsWorks for Puppet Enterprise	AWS OpsWorks for Chef Automate et OpsWorks pour Puppet Enterprise ont mis à jour les politiques gérées `AWSOpsWorksCMServiceRole` et `AWSOpsWorksCMInstanceProfileRole` y stockent désormais des secrets. AWS Secrets Manager	3 mai 2021

Mises à jour de AWS OpsWorks for Puppet Enterprise	La version du moteur d'un serveur OpsWorks pour Puppet Enterprise que vous créez dans la console est désormais 2019.8.5. À l'aide de l'API, vous pouvez spécifier la version `2019` ou le `2017` moment où vous créez un serveur Puppet Enterprise. L'`DescribeServers` API renvoie désormais un attribut appelé `PUPPET_API_CRL` dans ses résultats. Cet attribut contient une liste de révocation de certificats à usage interne.	28 avril 2021
AWS OpsWorks Stacks utilise une nouvelle politique gérée	AWS OpsWorks Stacks a modifié la politique gérée qui inclut les autorisations permettant d'effectuer toutes les actions dans AWS OpsWorks Stacks. La nouvelle politique est AWSOpsWorks_FullAccess. Pour plus d'informations sur les autorisations définies dans cette politique, consultez la section Exemples de politiques.	19 février 2021
Migrer AWS OpsWorks Stacks les stacks de EC2 -Classic vers un VPC	Une documentation a été ajoutée décrivant comment migrer une AWS OpsWorks Stacks pile de EC2 -Classic vers un VPC.	29 septembre 2020

Régénérez un kit de démarrage pour et AWS OpsWorks for Chef AutomateAWS OpsWorks for Puppet Enterprise	Une documentation a été ajoutée décrivant comment régénérer le kit de démarrage pour un AWS OpsWorks for Chef Automate ou un AWS OpsWorks for Puppet Enterprise serveur.	29 juillet 2020
AWS OpsWorks for Puppet Enterprise vous permet de créer un serveur qui utilise un domaine, un certificat et une clé privée personnalisés	Vous pouvez désormais créer un serveur OpsWorks pour Puppet Enterprise qui utilise un domaine, un certificat et une clé privée personnalisés. Vous pouvez mettre à jour un serveur Puppet Enterpris e existant pour utiliser un domaine personnalisé en créant un serveur à partir d'une sauvegarde d'un serveur existant.	17 avril 2020
AWS OpsWorks for Chef Automate et AWS OpsWorks for Puppet Enterprise supporte désormais le balisage dans la console	Vous pouvez désormais ajouter des balises à un AWS OpsWorks for Chef Automate serveur ou à un AWS OpsWorks for Puppet Enterprise maître, ou à des sauvegardes de serveurs, en utilisant le AWS Management Console ou le AWS CLI. Pour de plus amples informations, veuillez consulter Utiliser des balises (Chef) ou Utiliser des balises (Puppet).	26 février 2020

AWS OpsWorks for Chef Automate simplifie la mise à niveau des serveurs Chef Automate 1 existants vers Chef Automate 2	Vous pouvez mettre à niveau AWS OpsWorks for Chef Automate les serveurs éligibles exécutant Chef Automate 1 vers Chef Automate 2 en choisissant Démarrer la mise à niveau sur la page de détails de votre serveur dans la console, ou en exécutant l'action `StartMaintenance` API. Pour plus d'informations, voir Mettre à niveau un AWS OpsWorks for Chef Automate serveur vers Chef Automate 2.	24 janvier 2020
AWS OpsWorks for Chef Automate et AWS OpsWorks for Puppet Enterprise	Un nouveau chapitre sur la sécurité dans AWS OpsWorks CM (AWS OpsWorks for Chef Automate et AWS OpsWorks for Puppet Enterprise) a été ajouté au guide.	23 décembre 2019
AWS OpsWorks for Chef Automate et AWS OpsWorks for Puppet Enterprise soutenez le balisage	Vous pouvez désormais ajouter des balises à un AWS OpsWorks for Chef Automate serveur ou à un AWS OpsWorks for Puppet Enterprise maître, ou à des sauvegardes de serveurs, en utilisant le AWS CLI. AWS OpsWorks CM prend désormais en charge l'autorisation basée sur des balises.	18 décembre 2019

AWS OpsWorks for Chef Automate vous permet de créer un serveur qui utilise un domaine, un certificat et une clé privée personnalisés	Vous pouvez désormais créer un serveur AWS OpsWorks for Chef Automate 2.0 qui utilise un domaine, un certificat et une clé privée personnalisés. Vous pouvez mettre à jour un serveur Chef Automate 2.0 existant pour utiliser un domaine personnalisé en créant un serveur à partir d'une sauvegarde d'un serveur existant.	22 octobre 2019
AWS OpsWorks Stacks supporte désormais Ruby 2.6.1	AWS OpsWorks Stacks prend en charge les couches Ruby 2.6.1 on Rails App Server dans les stacks Chef 11.10.	2 mai 2019
AWS OpsWorks for Chef Automate prend désormais en charge Chef Automate 2.0	AWS OpsWorks for Chef Automate Les nouveaux serveurs exécuteront Chef Automate 2.0, qui inclut des mises à jour de Chef InSpec, de nouvelles fonctionnalités d'analyse de conformité et de création de rapports, ainsi que Chef Infra.	30 avril 2019
AWS OpsWorks for Chef Automate et AWS OpsWorks for Puppet Enterprise	Vous pouvez désormais l'utiliser AWS CloudFormation pour créer un AWS OpsWorks for Chef Automate serveur ou un serveur AWS OpsWorks for Puppet Enterprise principal.	24 janvier 2019

AWS OpsWorks Piles	AWS OpsWorks Stacks prend désormais en charge les instances exécutant Ubuntu 18.04 LTS dans les stacks Chef 12.	18 décembre 2018
AWS OpsWorks pour Puppet Enterprise	Ajout d'une procédure pour configurer une connexion SSH à un référentiel de contrôle qui utilise. CodeCommit	3 décembre 2018
AWS OpsWorks Piles	AWS OpsWorks Stacks prend désormais en charge les instances exécutant Amazon Linux 2 dans les stacks Chef 12.	15 novembre 2018
AWS OpsWorks Piles	AWS OpsWorks Stacks prend désormais en charge les instances exécutant Amazon Linux 2018.03 dans les stacks Chef 11.10.	23 octobre 2018
AWS OpsWorks Piles	AWS OpsWorks Stacks prend désormais en charge les instances exécutant Amazon Linux 2018.03 dans les stacks Chef 12.	23 août 2018
AWS OpsWorks for Chef Automate et OpsWorks pour Puppet Enterprise	OpsWorks for Puppet Enterprise a été mis à niveau vers PE 2018.1.2. AWS OpsWorks for Chef Automate a été mis à niveau vers Chef Automate 1.8.68.	29 juin 2018

- AWS OpsWorks for Chef Automate et OpsWorks pour la version de l'API Puppet Enterprise : 01/11/2016
- AWS OpsWorks Version de l'API Stacks : 08/03/2016
- Dernière mise à jour de la documentation : 2024-04-11

Mises à jour antérieures

Le tableau ci-après décrit des modifications importantes apportées dans chaque version du Guide de l'utilisateur AWS OpsWorks avant juin 2018.

Description	Date
AWS OpsWorks La version Stacks Chef pour les stacks basés sur Windows a été mise à jour vers la version 12.22 ; la version Ruby est désormais 2.3.6.	19 avril 2018
Nouvelles procédures pour créer un AWS OpsWorks for Chef Automate serveur ou un master OpsWorks pour Puppet Enterprise à l'aide du AWS CLI.	23 mars 2018
Version Chef Automate mise à jour vers 1.8 ; la configuration de Chef Compliance a été simplifiée avec l'ajout du `opsworks-audit` livre de recettes.	5 mars 2018
Ajout de la prise en charge AWS OpsWorks des événements Stacks dans Amazon CloudWatch Events.	20 février 2018
Ajout de la prise en charge des nouveaux types de volumes EBS dans AWS OpsWorks Stacks et d'une nouvelle API. DescribeOperatingS ystems	25 janvier 2018
OpsWorks pour Puppet Enterprise et prend AWS OpsWorks for Chef Automate désormais en charge la sélection de plusieurs groupes de sécurité lors de la création d'un serveur.	18 janvier 2018
Ajout du support pour AWS OpsWorks Stacks dans la région Europe (Paris).	19 décembre 2017

Description	Date
Ajout de la prise en charge de AWS OpsWorks for Chef Automate et OpsWorks pour Puppet Enterprise dans six régions supplémentaires, et de procédures supplémentaires pour créer des sauvegardes des AWS OpsWorks for Chef Automate serveurs Puppet Enterprise dans le. OpsWorks AWS Management Console	18 décembre 2017
Ajout du nouveau service et de la documentation OpsWorks pour Puppet Enterprise.	16 novembre 2017
Ajout de la prise en charge d'Amazon Linux 2017.09 à AWS OpsWorks Stacks.	7 novembre 2017
Ajout de la prise en charge de Chef Compliance à AWS OpsWorks for Chef Automate.	25 octobre 2017
Ajout du support pour Amazon Linux 2017.09 à. AWS OpsWorks for Chef Automate	9 octobre 2017
Ajout d'une rubrique sur la maintenance du système au AWS OpsWorks for Chef Automate chapitre.	28 juillet 2017
Ajout du support pour les tags dans AWS OpsWorks Stacks.	6 juin 2017
Intégration ajoutée avec CloudWatch Logs.	10 avril 2017
Ajout du nouveau AWS OpsWorks for Chef Automate service et de la nouvelle documentation.	1er décembre 2016
Ajout de la prise en charge du point de terminaison régional de la région USA Est (Ohio).	12 octobre 2016
Ajout de la prise en charge des piles et des instances qui exécutent le système d'exploitation Amazon Linux 2016.09.	30 septembre 2016
Ajout du support pour la région Asie-Pacifique (Séoul) et de neuf points de terminaison régionaux supplémentaires.	15 août 2016

Description	Date
Ajout de la prise en charge de Node.js 0.12.15 et de Ruby 2.3 dans les couches intégrées.	6 juillet 2016
Ajout du support pour la région Asie-Pacifique (Mumbai).	28 juin 2016
Ajout de la prise en charge des piles et des instances qui exécutent le système d'exploitation CentOS 7.	22 juin 2016
Ajout d'une description détaillée CodePipeline et de l'intégration de AWS OpsWorks Stacks.	2 juin 2016
Ajout de la prise en charge des piles et des instances qui exécutent le système d'exploitation Ubuntu 16.04 LTS.	1 juin 2016
Ajout de la prise en charge de Linux Chef 12 et des documents associés.	3 décembre 2015
Ajout de la procédure Node.js à la mise en route.	14 juillet 2015
Ajout de deux exemples de livres de recettes dans les bases des livres de recettes.	14 juillet 2015
Ajout de la prise en charge de la gestion des versions de l'agent.	23 juin 2015
Ajout de la prise en charge de la gestion de la version de l'agent.	24 juin 2015
Ajout du support pour Windows personnalisé AMIs.	22 juin 2015
Ajout de trois nouvelles rubriques sur les bonnes pratiques.	11 juin 2015
Ajout de la prise en charge des piles Windows.	18 mai 2015
Ajout d'un chapitre sur les bonnes pratiques.	15 décembre 2014
Ajout de la prise en charge du drainage des connexions Elastic Load Balancing et des délais d'arrêt personnalisés.	15 décembre 2014
Ajout de la prise en charge de l'enregistrement des instances créées en dehors d'AWS OpsWorks Stacks.	9 décembre 2014

Description	Date
Ajout de la prise en charge d'Amazon SWF.	4 septembre 2014
Ajout de la prise en charge de l'association des variables d'environ nement aux applications et aux bases des livres de recettes étendus.	16 juillet 2014
Ajout du didacticiel sur les bases des livres de recettes pour présenter l'implémentation des livres de recettes.	16 juillet 2014
Ajout du support pour CloudTrail.	4 juin 2014
Ajout de la prise en charge d'Amazon RDS.	14 mai 2014
Ajout de la prise en charge de Chef 11.10 et de Berkshelf.	27 mars 2014
Ajout de la prise en charge des volumes Amazon EBS PIOPS.	16 décembre 2013
Ajout d'autorisations basées sur les ressources.	5 décembre 2013
Ajout de la gestion des ressources.	7 octobre 2013
Ajout du support pour VPCs.	29 août 2013
Ajout de la prise en charge de la personnalisation AMIs et de Chef 11.4.	24 juillet 2013
Ajout de la prise en charge par la console de plusieurs couches par instance.	1 juillet 2013
Ajout de la prise en charge des instances soutenues par Amazon EBS, d'Elastic Load Balancing et de la surveillance Amazon CloudWatch.	14 mai 2013
Première publication du guide de l'utilisateur d'AWS OpsWorks Stacks.	18 février 2013

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.